Learning of Balance Controller Considering Changes in Body State for Musculoskeletal Humanoids

Kento Kawaharazuka[1], Yoshimoto Ribayashi[1], Akihiro Miki[1], Yasunori Toshimitsu[1],
Temma Suzuki[1], Kei Okada[1], and Masayuki Inaba[1]

Abstract— The musculoskeletal humanoid is difficult to modelize due to the flexibility and redundancy of its body, whose state can change over time, and so balance control of its legs is challenging. There are some cases where ordinary PID controls may cause instability. In this study, to solve these problems, we propose a method of learning a correlation model among the joint angle, muscle tension, and muscle length of the ankle and the zero moment point to perform balance control. In addition, information on the changing body state is embedded in the model using parametric bias, and the model estimates and adapts to the current body state by learning this information online. This makes it possible to adapt to changes in upper body posture that are not directly taken into account in the model, since it is difficult to learn the complete dynamics of the whole body considering the amount of data and computation. The model can also adapt to changes in body state, such as the change in footwear and change in the joint origin due to recalibration. The effectiveness of this method is verified by a simulation and by using an actual musculoskeletal humanoid, Musashi.

I. INTRODUCTION

A variety of musculoskeletal humanoids have been developed so far [1]–[3]. Due to the flexibility and redundancy of their bodies, all of them are very difficult to control in the same way as ordinary axis-driven robots. Various learning-based control methods have been proposed for them. In [4], a pedaling operation is acquired by self-repetitive learning. [5] proposes to control the upper body reaching motion using reinforcement learning. In [6], for a relatively simple system with one or two joints, the relationship among joints, muscles, and tasks is trained, and a robot is controlled using the trained neural network. In [7], [8], for a more complex system, the relationship among joint angle, muscle tension, and muscle length is modelized by a neural network, which is trained and applied mainly to upper body control and state estimation. [9] has succeeded in recognizing grasped objects and stabilizing tool grasping by learning the dynamics of a musculoskeletal hand. These methods have made it possible for complex musculoskeletal robots to acquire the ability to control themselves autonomously.

On the other hand, the balance control of these robots is still difficult. In the case of balance control, data collection itself is difficult, because data must be acquired while the robot is in a balanced state. Therefore, none of the studies

[1] The authors are with the Department of Mechano-Informatics, Graduate School of Information Science and Technology, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo, 113-8656, Japan. [kawaharazuka, ribayashi, miki, toshimitsu, t-suzuki, k-okada, inaba]@jsk.t.u-tokyo.ac.jp
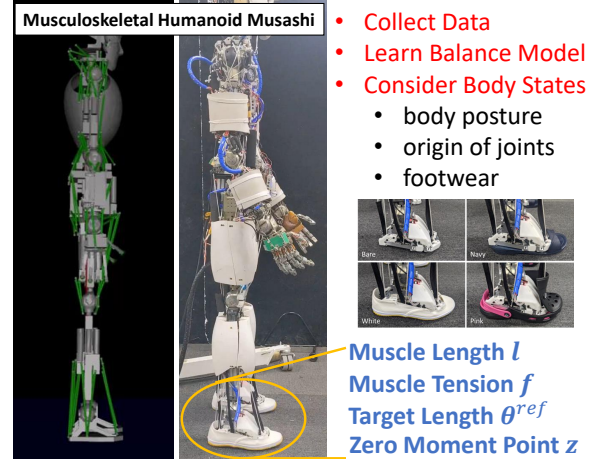
Fig. 1. The concept of this study.

described so far deals with balancing. Although a simple balance control using PID has been implemented [10], it is difficult to say that its balance has improved because the convergence of zero moment point has not been evaluated, and the success rate of the step-out experiment is extremely low. In addition, there is usually a strong human parameterization according to the structure of the robot, and the robot does not acquire the balance control autonomously. Exempting musculoskeletal humanoids, methods to solve this problem have been developed by using real2sim [11] and sim2real [12] to perform reinforcement learning in simulation environments. A safe learning with dynamics balancing models [13] and locomotion generation with learning by cheating [14] has also been developed. In addition, for quadruped robots, where balance is relatively easy to handle, running motion is generated only by learning on the actual robot [15]. On the other hand, it is very difficult to construct a model of a complex body such as the musculoskeletal humanoid in a simulation, and also it is challenging for the actual bipedal robot to collect data for model learning while maintaining balance. Even if we can construct a simulation, it is difficult to transfer the simulation model to the actual robot because of the large differences in muscle elongation, friction, muscle paths, etc. Therefore, it is desirable to acquire balance control autonomously by learning the relationships among various sensor values only in the actual robot. Also, it is necessary to solve the problem of the difficult data collection in the actual robot.

In addition, there is a problem that there are many changes in dynamics of the body that cannot be directly represented in
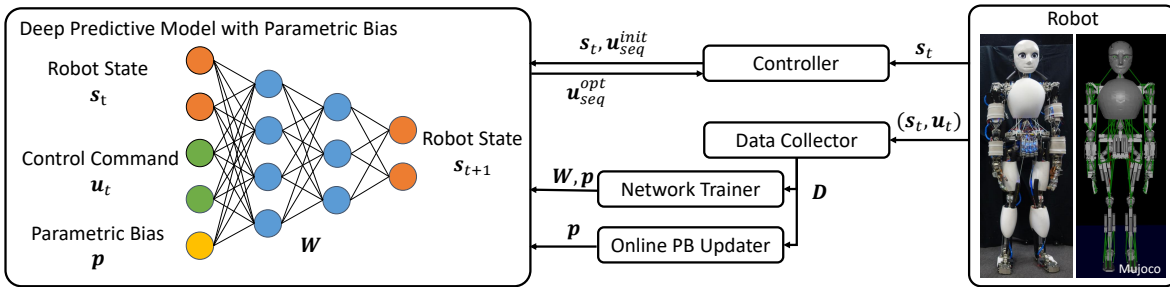
Fig. 2. The overall system of balance control for musculoskeletal humanoids using a deep predictive model with parametric bias.

the model due to changes in the current body state. First, as it is difficult to learn the dynamics of all the sensor values of the whole body since it requires a huge amount of data, we will learn only some of the dynamics. In other words, if we learn the dynamics model among the ankle joints, muscles, and zero moment point, the changes in dynamics caused by the changes in upper body posture, etc., which are not included in the model, cannot be taken into account. Second, changes in the origin of muscles and joints due to irreproducible calibration, which is characteristic to musculoskeletal structures that usually do not have joint angle sensors, cannot be taken into account. In addition, if the shoes worn by the humanoid change, the dynamics will change significantly, which must also be taken into account. These changes in dynamics should be handled as disturbances or embedded as some low-dimensional parameters, and the balance control should be adapted to them. However, the former method that handles dynamics changes as disturbances usually requires accurate body models and prior knowledge of the distribution of disturbances. Also, since humans do not treat changes in the shoes that they are wearing as disturbances, but rather their walking style changes in response to the shoes, we believe that a new control method that incorporates this phenomenon is important. In this study, we apply the mechanism of parametric bias [16] to this problem. Parametric bias is a bias parameter that allows multiple attractor dynamics to be implicitly embedded in a neural network, and has been mainly used in the context of imitation learning. In this study, we use this variable to learn a predictive model of various sensors for body balance, and embed the information of the changes in body state described so far in parametric bias. By learning this predictive model and estimating the changes in body state online, it is possible to perform balance control while adapting to the current body state (Fig. 1). Note that, although parametric bias is also used in [9], [17], this study examines data collection methods and application to changes in body state including wearing shoes for balance control of a flexible body.

The purpose of this study is to develop a balance control system for humanoids with complex and flexible bodies that are difficult to modelize and whose body states change over time. Therefore, we develop a balance control system for musculoskeletal humanoids using a Deep Predictive Model with Parametric Bias (DPMPB). This enables not only autonomous learning of balance control but also adaptive control to changes in upper body posture and shoes, which

are not directly included in the model. The contributions of this study are summarized as follows.

- Data collection for balance control in the actual musculoskeletal humanoid
- Embedding of changes in body state including wearing shoes into the model using parametric bias
- Online adaptation to the current body state and balance control using DPMPB

This method is applied to a simulation and an actual musculoskeletal humanoid, Musashi [3], to confirm its effectiveness.

II. Balance Control of Musculoskeletal Humanoids Using Deep Predictive Model with Parametric Bias

The overall system of balance control using DPMPB is shown in Fig. 2.

A. Network Structure of DPMPB

The network structure of DPMPB is shown below,

$$s_{t+1} = h(s_t, u_t, p) \tag{1}$$

where t is the current time step, s is the sensor state, u is the control input, p is parametric bias, and h is the network of DPMPB. In this study, for the balance control in the musculoskeletal humanoid, we directly deal with the state of joints and muscles related to the ankles, while the posture of the upper body is implicitly handled by parametric bias. Therefore, we set $s = \begin{pmatrix} z^T & f^T & l^T \end{pmatrix}^T$ and $u = \theta^{ref}$. Here, z is zero moment point (ZMP), $\{f, l\}$ is {muscle tension, muscle length} regarding the ankles of both legs, and θ^{ref} is the target joint angle of the ankles. Note that z is 2-dimensional (z_x for x-direction and z_y for y-direction), and the dimension of $\{f, l\}$ depends on the robot configuration. Although θ^{ref} can have roll and pitch angles for both legs, we assume the angles for both legs to be the same and θ^{ref} to be 1-dimensional only for the pitch axis in this study, for simplicity. Parametric bias p is an input variable that can embed implicit differences in dynamics, and in this study, by collecting data while changing the body states (the posture of the upper body, calibration, shoes, etc.), this information is self-organized in p. h is a predictive model that represents the state transition of s by u, and the dynamics of the model can be modified by changing p.

In this study, DPMPB has 10 layers, which consist of four FC layers (fully-connected layers), two LSTM layers (long short-term memory layers), and four FC layers. The number

of units is set to $\{N_s + N_u + N_p, 200, 100, 30, 30$ (number of units in LSTM), 30 (number of units in LSTM), 30, 100, 200, $N_s\}$ (where $N_{\{s,u,p\}}$ is the dimensionality of $\{s, u, p\}$). The activation function is hyperbolic tangent and the update rule is Adam [18]. We also set p to be two-dimensional and the execution period of Eq. 1 is 5 Hz. The dimension of p should be slightly smaller than the expected changes in the body state, because too small a dimensionality will not represent the change in dynamics properly, and too large a dimensionality will make self-organization of p difficult.

B. Data Collection

In order to learn balance control, some technique is needed in the method of data collection. If we simply move the ankles randomly, the robot will quickly fall down and it is difficult to collect useful data for balance control. In this study, θ^{ref} is varied by repeating the following process at each step,

$$c \leftarrow c + 1 \tag{2}$$
$$d \leftarrow d + C_{diff} \tag{3}$$
$$\theta^{ref} \leftarrow \theta^{ref} + |\sin(\pi \frac{c}{N_{cnt}})|\text{Random}(-d, d) \tag{4}$$
$$\theta^{ref} \leftarrow \max(\theta_{min}, \min(\theta^{ref}, \theta_{max})) \tag{5}$$

where c is the time count (starting from $c = 0$), d is the maximum displacement of θ^{ref} (starting from $d = C_{diff}^{init}$), and Random(a, b) is a random value in the range of $[a, b]$. Also, $\theta_{\{min, max\}}$ is {minimum, maximum} value of θ^{ref}, and $\{N_{cnt}, C_{diff}, C_{diff}^{init}\}$ is a constant that determines the behavior of data collection. It collects data while gradually increasing the maximum value of the displacement of θ^{ref} with d. This is important because if the displacement is too large at the beginning, it will quickly fall down and we will not be able to collect data for a long time. Also, by periodically decreasing or increasing the change of θ^{ref} with c, we can collect various data. Since the best state for balance control is a stable stationary state, if we do not collect data for stationary states where the displacement of θ^{ref} is small, oscillatory motions will be generated during balance control. Finally, θ^{ref} is clipped by the set minimum and maximum values.

In the experiments, in addition to the data collection by Eq. 4 (Proposed Collection), the following two types of data collection are compared,

$$\theta^{ref} \leftarrow \theta^{ref} + \text{Random}(-d, d) \tag{6}$$
$$\theta^{ref} \leftarrow \theta^{ref} + \text{Random}(-1.0, 1.0) \tag{7}$$

where we call Eq. 6 Gradual Collection and Eq. 7 Random Collection. Gradual Collection is a collection method excluding the periodic change of θ^{ref} from Proposed Collection, and Random Collection is a collection method excluding the gradual increase of θ^{ref} from Gradual Collection.

In this study, we set $N_{cnt} = 50$, $C_{diff} = 0.002$ [rad], $C_{diff}^{init} = 0.1$ [rad], $\theta_{min} = -1.0$ [rad], and $\theta_{max} = 1.0$ [rad]. Since the body is very difficult to modelize, some experimental tuning of these coefficients is necessary.

C. Training of DPMPB

Using the obtained data D, DPMPB is trained. In this process, we can implicitly embed the information of body state into parametric bias by collecting data while changing the body state. In order to allow each time-series data transition with different dynamics to be represented by a single model, the differences in the dynamics is self-organized in a low-dimensional space of p. It can be regarded as a weakly supervised learning, in which only weak labels are given, i.e., whether or not the body state is the same for each data.

The data collected in the same body state k is represented as $D_k = \{(s_1, u_1), (s_2, u_2), \cdots, (s_{T_k}, u_{T_k})\}$ ($1 \leq k \leq K$, where K is the total number of body states and T_k is the number of motion steps for the body state k), and the data used for training $D_{train} = \{(D_1, p_1), (D_2, p_2), \cdots, (D_K, p_K)\}$ is constructed. Here, p_k is the parametric bias that represents the dynamics in the body state k, which is a common variable for that state and a different variable for another state. We use D_{train} to train the DPMPB. In an ordinary learning process, only the network weight W is updated, but here, W and p_k for each state are updated simultaneously. In this way, p_k embeds the difference of dynamics in each body state. In the learning process, the mean squared error is used as the loss function, and p_k is optimized with all initial values set to $\mathbf{0}$.

D. Online Update of Parametric Bias

Using the data D obtained in the current body state, we update parametric bias online. If the network weight W is updated, DPMPB may overfit to the data, but if only the low-dimensional parametric bias p is updated, no overfitting occurs and life-long update is possible. Note that it has been experimented in [19] that fine tuning of only W without using p cannot deal with various body states, though this is a study on a static motion model. This online learning allows us to obtain a model that is always adapted to the current body state.

Let the number of data obtained be N_{data}^{online}, and start online learning when the number of data exceeds N_{thre}^{online}. For each new data, we fix W and update only p by setting the number of batches as N_{batch}^{online}, the number of epochs as N_{epoch}^{online}, and the update rule as MomentumSGD. Data exceeding N_{max}^{online} are deleted from the oldest ones.

In this study, we set $N_{\{thre, max\}}^{online} = 50$, $N_{batch}^{online} = N_{max}^{online}$, and $N_{epoch}^{online} = 1$.

E. Balance Control using DPMPB

We describe a control method using DPMPB. Here, we consider optimizing u from the loss function for s and u. First, we give the initial value u_{seq}^{init} for the time-series control input $u_{seq} = u_{[t:t+N_{step}-1]}$ (N_{step} represents the number of DPMPB expansions, or control horizon). Let u_{seq}^{opt} be u to be optimized, and repeat the following calculation at the time
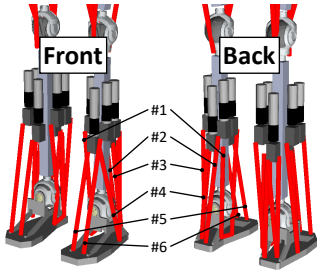
Fig. 3. The muscle arrangement of the musculoskeletal humanoid Musashi.

step t to obtain the optimal $\boldsymbol{u}_t^{opt}$,

$$\boldsymbol{s}_{seq}^{pred} = \boldsymbol{h}_{expand}(\boldsymbol{s}_t, \boldsymbol{u}_{seq}^{opt}) \tag{8}$$

$$L = \boldsymbol{h}_{loss}(\boldsymbol{s}_{seq}^{pred}, \boldsymbol{u}_{seq}^{opt}) \tag{9}$$

$$\boldsymbol{u}_{seq}^{opt} \leftarrow \boldsymbol{u}_{seq}^{opt} - \gamma \partial L / \partial \boldsymbol{u}_{seq}^{opt} \tag{10}$$

where $\boldsymbol{s}_{seq}^{pred}$ is the predicted $\boldsymbol{s}_{[t+1:t+N_{step}]}$, $\boldsymbol{h}_{expand}$ is the function of $\boldsymbol{h}$ expanded N_{step} times, $\boldsymbol{h}_{loss}$ is the loss function, and γ is the learning rate. Thus, the future $\boldsymbol{s}$ is predicted from the current sensor state $\boldsymbol{s}_t$ by $\boldsymbol{u}_{seq}^{opt}$, and $\boldsymbol{u}_{seq}^{opt}$ is optimized by using the backpropagation and gradient descent methods to minimize the loss function.

In this process, we set $\boldsymbol{u}_{seq}^{init}$ as $\boldsymbol{u}_{\{t+1,\cdots,t+N_{step}-1,t+N_{step}-1\}}^{prev}$ by using $\boldsymbol{u}_{[t:t+N_{step}-1]}^{prev}$, which is $\boldsymbol{u}_{seq}$ optimized in the previous step, shifting the time by one, and replicating the last term. By using the previous optimization result, faster convergence can be obtained. For γ, we prepare $N_{batch}^{control}$ number of γ, which are exponentially divided $[0, \gamma_{max}]$, and after running Eq. 10 on each γ, we calculate Eq. 9 and select the $\boldsymbol{u}_{seq}^{opt}$ with the lowest loss, repeating the process $N_{epoch}^{control}$ times. Faster convergence can be obtained by trying various γ and always choosing the best learning rate.

Here, we consider the loss function. In this study, we set $\boldsymbol{h}_{loss}$ as follows,

$$\begin{aligned} \boldsymbol{h}_{loss}(\boldsymbol{s}_{seq}^{pred}, \boldsymbol{u}_{seq}^{opt}) = &\|\boldsymbol{z}_{seq}^{pred} - \boldsymbol{z}_{seq}^{ref}\|_2 \\ &+ C_f \|\boldsymbol{f}_{[3:N_{step}]}^{pred} - \boldsymbol{f}_{[2:N_{step}-1]}^{pred}\|_2 \\ &+ C_l \|\boldsymbol{l}_{[3:N_{step}]}^{pred} - \boldsymbol{l}_{[2:N_{step}-1]}^{pred}\|_2 \\ &+ C_u \|\boldsymbol{u}_{seq}^{opt}\|_2 \end{aligned} \tag{11}$$

where $\{\boldsymbol{z}, \boldsymbol{f}, \boldsymbol{l}\}_{seq}^{pred}$ is the value of $\{\boldsymbol{z}, \boldsymbol{f}, \boldsymbol{l}\}$ in $\boldsymbol{s}_{seq}^{pred}$, $\boldsymbol{z}_{seq}^{ref}$ is the value obtained by arranging N_{step} target values of $\boldsymbol{z}$, and $C_{\{f,l,u\}}$ is the constant weight for each loss. Thus, the loss is a summary of the realization of the target value for $\boldsymbol{z}$, the minimization of the change in $\boldsymbol{f}$, the minimization of the change in $\boldsymbol{l}$, and the minimization of $\boldsymbol{u}$. Note that $C_{\{f,l,u\}}$ is varied for each experiment.

In this study, we set $N_{step} = 6$, $N_{batch}^{control} = 10$, $N_{epoch}^{control} = 3$, and $\gamma_{max} = 0.1$.

III. EXPERIMENTS

A. Experimental Setup

In this study, we conduct experiments using the musculoskeletal humanoid Musashi (Fig. 1) [3]. Musashi has redundant 74 muscles including 4 polyarticular muscles in
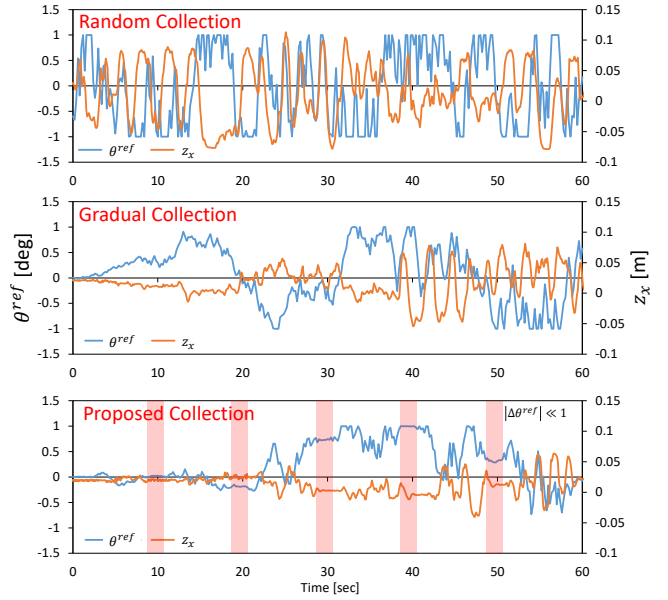

Fig. 4. Simulation experiment: the transition of θ^{ref} and z_x when conducting Random, Gradual, and Proposed Collections.

its body and 34 over-actuated joints excluding fingers. In this study, Musashi basically moves only the pitch joint of the ankle and controls balance in an upright posture except for in some upper body postures. As shown in Fig. 3, there are six muscles for each ankle joint, and the dimensionality of $\{\boldsymbol{f}, \boldsymbol{l}\}$ for both legs is 12. ZMP is calculated from 12 loadcells distributed in the foot. [20] is used to convert the target joint angle to target muscle length, assuming the target muscle tension to be constant at 100 [N]. Note that the learning of [20] does not perfectly realize the target joint angle, and there are some errors due to muscle friction and other factors. For simulation, we use Mujoco [21].

B. Simulation Experiment

1) Training of DPMPB: In this experiment, we handle the pitch angle of the spine joint θ_{s-p}, and the offset of the pitch angle of the ankle joint θ_{a-p}^{offset} representing irreproducible calibration. First, we collect data while changing the body state to nine combinations of $\theta_{s-p} = \{-5.0, 0.0, 5.0\}$ [deg] and $\theta_{a-p}^{offset} = \{-5.0, 0.0, 5.0\}$ [deg]. For each body state, we obtain data for 300 time steps. Here, the transitions of z_x and θ^{ref} are shown in Fig. 4 when using Proposed, Gradual, or Random Collection in Section II-B. In Random Collection, z_x and θ^{ref} continue to change significantly. In Gradual Collection, the range of change in z_x and θ^{ref} gradually increases. On the other hand, in Proposed Collection, in addition to the characteristics of Gradual Collection, θ^{ref} alternates between violent and slow motions, and a variety of data is collected. We train DPMPB using the data of 2700 time steps. For each data collection method, the obtained parametric bias is converted by Principle Component Analysis (PCA) and plotted on a two-dimensional plane as shown in Fig. 5. In Proposed Collection, we can see that the space of PB is self-organized according to the size of the body state parameters, though the parameters related to the body state are not
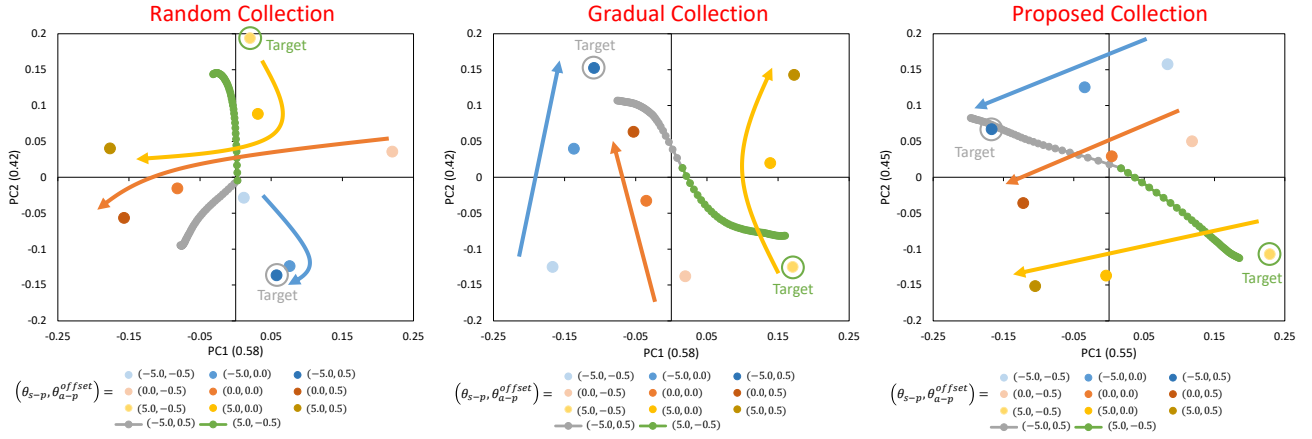
Fig. 5. Simulation experiment: the arrangement of parametric bias when training DPMPB using the data collected with Random, Gradual, and Proposed Collection, and the trajectories of parametric bias when running online learning by setting $(\theta_{s-p}, \theta_{a-p}^{offset}) = \{(-5.0, 0.5), (5.0, -0.5)\}$.
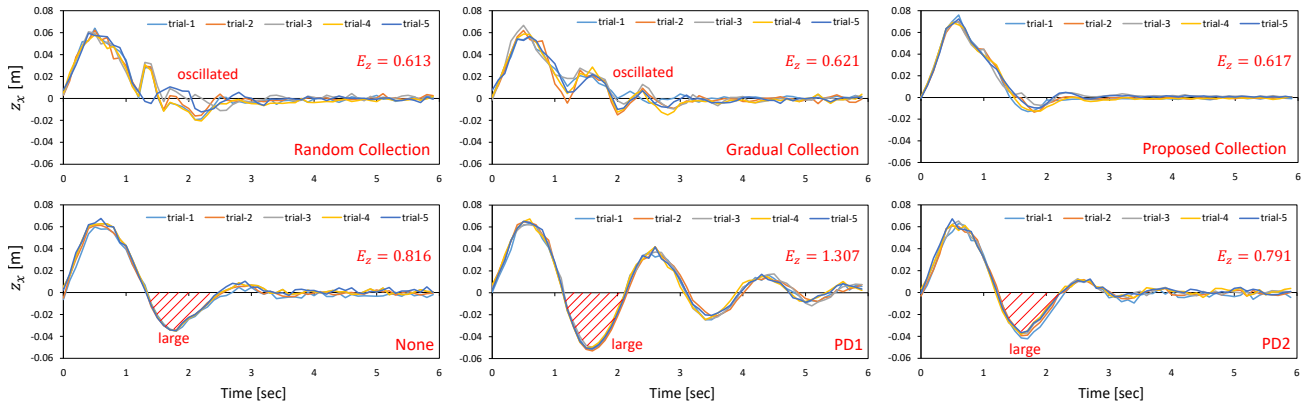

Fig. 6. Simulation experiment: the transitions of z_x after applying 30 N force to the chest link for 0.2 seconds, when the balance control using DPMPB trained with the data collected by Random, Gradual, or Proposed Collection is performed, when no balance control is performed (None), or when simple PD controls with different gains are performed (PD1, PD2).
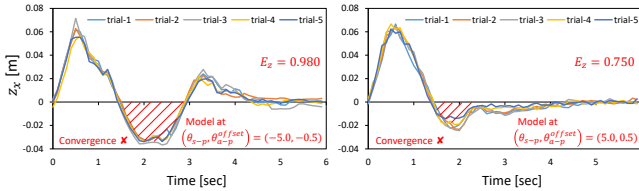

Fig. 7. Simulation experiment: the transitions of z_x after external force of 30 N to the chest link for 0.2 seconds when running the balance control using DPMPB trained with the data collected by Proposed Collection and parametric bias trained at $(\theta_{s-p}, \theta_{a-p}^{offset}) = \{(-5.0, -0.5), (5.0, 0.5)\}$.

directly given as data. In other words, even in the case where the parameters of the body state are not directly available, such as in the recalibration of the actual robot, it is possible to structure the information in the space of PB. On the other hand, Gradual Collection shows a more distorted space of PBs than Proposed Collection. As for Random Collection, the space of PB is even more distorted than that of Gradual Collection.

2) Online Update of Parametric Bias: Starting from the state of $\boldsymbol{p} = \boldsymbol{0}$, we examine how $\boldsymbol{p}$ transitions when the online update of PB is performed at the same time as when the body is moved the same way as in the data collection. The

trajectory of $\boldsymbol{p}$ when $(\theta_{s-p}, \theta_{a-p}^{offset}) = \{(-5.0, 0.5), (5.0, -0.5)\}$ is shown in Fig. 5. Note that the trajectories for 45 online learning steps are shown. It can be seen that the current $\boldsymbol{p}$ is gradually approaching the $\boldsymbol{p}$ previously trained in the same body state as the current state. In other words, it is possible to correctly recognize the body state by searching the space of $\boldsymbol{p}$. In addition, the accuracy of the recognition increases in the order of Random < Gradual < Proposed Collection.

3) Balance Control Using DPMPB: In this experiment, we set $(\theta_{s-p}, \theta_{a-p}^{offset}) = (0.0, 0.0)$, and the transition of z_x after applying an external force of 30N to the waist link for 0.2 seconds is examined five times for 6 seconds each. For z_x, offsets are removed to align the origins of the plots, and the average of the sum of $|z_x|$ for 6 seconds (30 steps) is shown as E_z. Unless otherwise stated, the constant weight for the loss function is set to $(C_f, C_l, C_u) = (0, 30, 3)$, and PB is the value obtained when $(\theta_{s-p}, \theta_{a-p}^{offset}) = (0.0, 0.0)$.

First, we show the results for the cases of balance control using the models obtained for Random, Gradual, and Proposed Collections, no control (None), and PD control (PD), in Fig. 6. As examples of PD controls, we show the cases of $(K_P, K_D) = (0.1, 0.1)$ (PD1) and $(K_P, K_D) = (0.03, 0.1)$ (PD2), though any PD setting would have worked to prevent
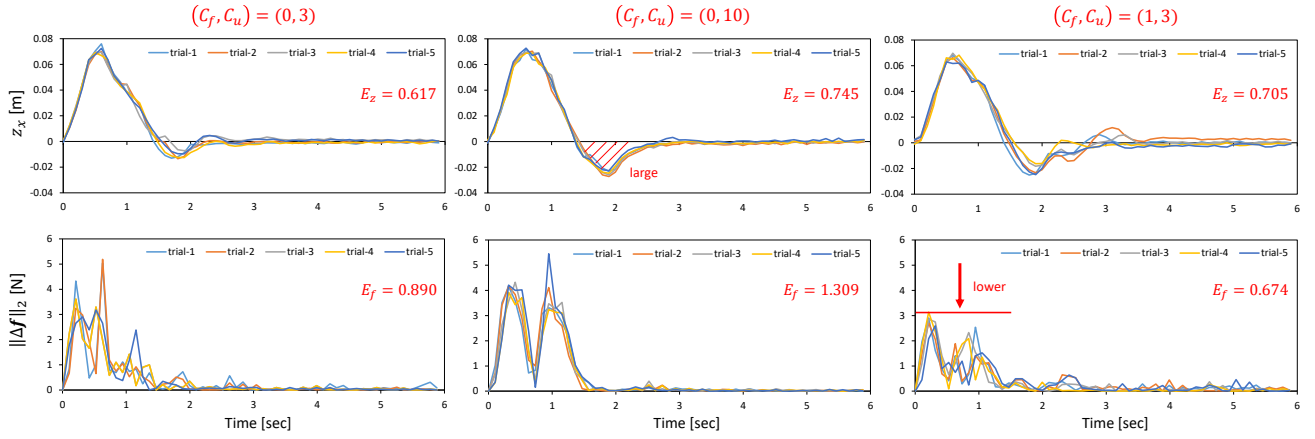
Fig. 8. Simulation experiment: the transitions of z_x and $\|\Delta f\|_2$ when running the proposed balance control with $(C_f, C_u) = \{(0,3), (0,10), (1,3)\}$ after external force of 30 N is applied to the chest link for 0.2 seconds.

the convergence of z_x ($K_{\{P,D\}}$ is the gain for PD control). Note that PD2 is the best controller tuned manually but it cannot be denied that tuning methods such as [22] may somewhat improve the results. It can be seen that the models using Random Collection and Gradual Collection do not change E_f significantly compared to the model using Proposed Collection, but z_x becomes oscillatory. In the case of None, E_z is larger than when using the control of this study, and x_z swings once in the positive direction and then again in the negative direction. On the other hand, in the case of our control, E_z does not swing much in the negative direction and converges faster. In the case of PD, even if the gain is changed, the convergence becomes worse than None in most cases.

Next, the results for the case where the model obtained in Proposed Collection is used and PB is the value obtained when $(\theta_{s-p}, \theta_{a-p}^{offset}) = \{(-5.0, -0.5), (5.0, 0.5)\}$ are shown in Fig. 7. For both cases, we can see that the error is much larger than that when using the correct PB obtained at $(\theta_{s-p}, \theta_{a-p}^{offset}) = (0.0, 0.0)$.

Finally, for the case of the model obtained in Proposed Collection, $C_l = 30$ is fixed and the parameters of the balance control are varied as $(C_f, C_u) = \{(0,3), (1,3), (0,10)\}$. The results are shown in Fig. 8. The transition of the norm $\|\Delta f\|_2$ of the time variation of the muscle tension from the previous step is also shown here, and the root mean square of the values is denoted by E_f. The upper left figure of Fig. 8 is the same graph as Proposed Collection of Fig. 6. It can be seen that changing C_u from 3 to 10 suppresses the movement of $u = \theta^{ref}$, so that the movement of z_x approaches None in Fig. 6. It can also be seen that when C_f is changed from 0 to 1, the peak of $\|\Delta f\|_2$ subsides and E_f becomes 0.674, which is smaller than in the case of $C_f = 0$.

C. Actual Robot Experiment

1) Training of DPMPB: In this experiment, we handle changes in the body state, such as which shoes to wear among Fig. 9 {Hard-Bare, Hard-White, Soft-Pink, Soft-Navy}, and the posture of the upper body {M, Z, P, U} (M is at $\theta_{s-p} = -5$, Z is at $\theta_{s-p} = 0$, P is at $\theta_{s-p} = 5$, and U
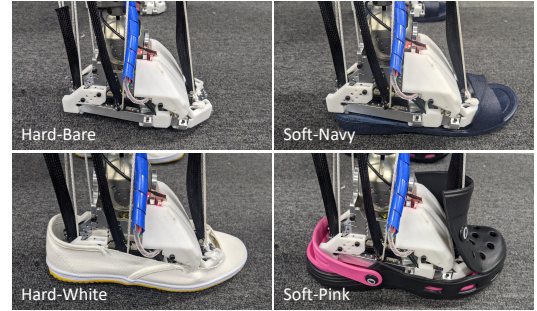

Fig. 9. Various shoes used for the actual robot experiment as temporal body changes.

is at $\theta_{e-p} = -60$ [deg], where θ_{s-p} is the pitch angle of the spine joint and θ_{e-p} is the pitch angle of the elbow joint). First, we collect the data while changing the body state into 12 different types, by changing shoes to {Hard-Bare, Soft-Pink, Soft-Navy} and upper body posture to {M, Z, P, U} (referred to as Hard-Bare/U or Soft-Navy/Z). For each body state, we obtain data for 300 steps. Here, we only collect data by Eq. 4, and the trained balance control is denoted as Proposed, while the case without any control is denoted as None. Parametric bias obtained by training DPMPB with these data is plotted on a two-dimensional plane by applying PCA to it, as shown in Fig. 10. We can see that the space of PB is roughly structured for Soft-Navy, Soft-Pink, and Hard-Bare. It can also be seen that the upper body postures of P and U have similar dynamics in the sense that the robot leans forward, and that the PBs of P and U are relatively close to each other. For this model, fine tuning from DPMPB trained in simulation does not reduce the loss much because the dynamics is very different.

2) Online Update of Parametric Bias: We start with p in Hard-Bare/U and examine how p transitions when the online update of PB is executed at the same time as the body is moved as in the data collection. The trajectories of p when the current body states are Soft-Pink/U, Soft-Navy/Z, and Hard-White/Z are shown in Fig. 10. For Soft-Pink/U and Soft-Navy/Z, we can see that the current p gradually approaches the p trained in the same body state as the current one. Thus, it is possible to correctly recognize the body state
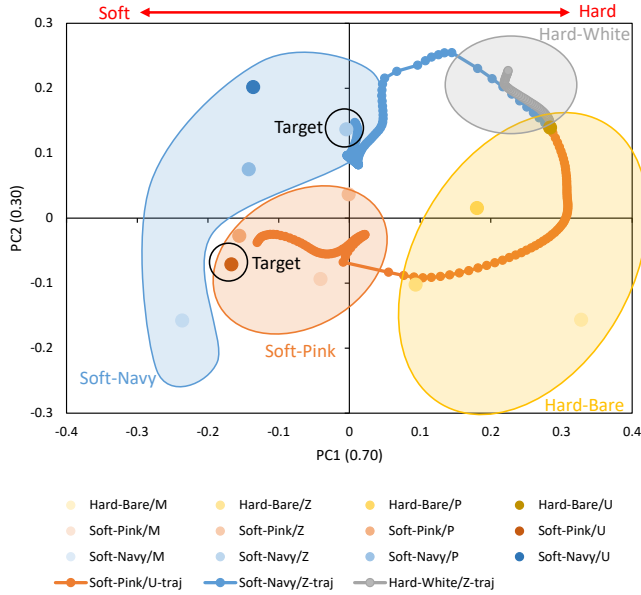
Fig. 10. Actual robot experiment: the arrangement of parametric bias when training DPMPB using the data collected with Proposed Collection, and the trajectories of parametric bias when running online learning by setting the body state to Soft-Pink/U, Soft-Navy/Z, or Hard-White/Z.
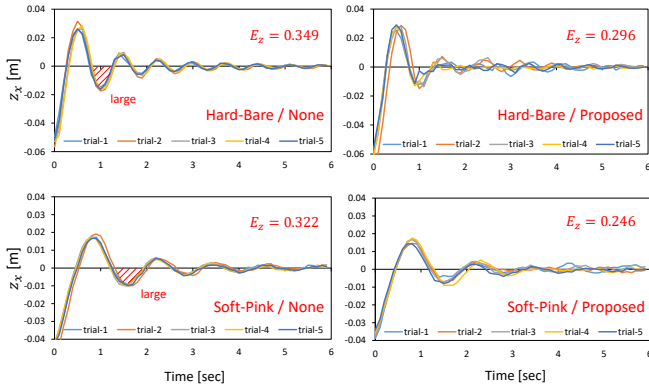

Fig. 11. Actual robot experiment: the transitions of z_x when 15 N (for Hard-Bare) or 10 N (for Soft-Pink) of external force is applied to the chest link and released, while the balance control using DPMPB is performed (Proposed), or while no control is performed (None).

by searching the space of p. Although Hard-White is not included in the training data, it is placed near the upper part of Hard-Bare as a result of online learning. Shoes have various parameters such as shape, friction, and softness, but the soles of Hard-White and Hard-Bare, Soft-Pink and Soft-Navy are similar in hardness.

3) Balance Control Using DPMPB: In this experiment, the upper body posture is Z, and the transition of z_x after applying a certain force to the waist link (15 N for Hard-Bare and 10 N for Soft-Pink) and then releasing it is examined five times for 6 seconds. The results of the balance control for Proposed and None are shown in Fig. 11. For z_x, offsets are removed to align the origins of the plots, and the average of the sum of $|z_x|$ for 6 seconds (30 steps) is shown as E_z. For PB, we use the values obtained while training for each body state (Hard-Bare/Z or Soft-Pink/Z). For Hard-Bare/Z, we set $(C_f, C_l, C_u) = (0, 30, 3)$, and for Soft-Pink/Z,

$(C_f, C_l, C_u) = (0, 3, 1)$. Although the effect is not as large as in the simulation, it can be seen that the convergence after the external force is faster in Proposed than in None. In fact, for Hard-Bare, $E_z = 0.349$ for None and $E_z = 0.296$ for Proposed, and for Soft-Pink, $E_z = 0.322$ for None and $E_z = 0.246$ for Proposed, indicating that Proposed has less error.

IV. Discussion

We discuss the experimental results of this study. First, the simulation results show that the parameters of the dynamics not explicitly given as values are embedded in parametric bias by learning the DPMPB. This arrangement of PB is self-organized nicely as the collected data has more diverse time-series changes, and PB can be updated online to adapt to the current dynamics. In addition, it can be seen that learning from data with various time series changes makes the balance control more accurate and the convergence of the response to external forces faster. In the case of no control or PID control, the convergence may be slow or divergence may occur, but our method enables the robot to stand upright stably and immediately after external force. On the other hand, when PB is not adapted to the current body state, the balance control may not work well due to the difference between the predicted dynamics and the actual dynamics. By changing the weights in the loss function, this balance control can simultaneously execute other objectives, such as reducing the control input and suppressing the changes in muscle length and tension.

Second, in the actual robot experiment, we handled the difference in dynamics of shoes, which is difficult to be given as values explicitly by humans. The trained PBs are grouped according to the type of shoe, and it is possible to estimate the type of shoe that the robot is currently wearing based on the current motion and understand the dynamics. The space of PB is constructed to reflect the nearness and remoteness of the dynamics that could be generalized to shoes that are not used for training. In addition, upper body postures such as the elbow and hip angles can be treated in the same variable of PB, in the form of changes in the dynamics of the lower body. The balance control shows some performance, and the convergence of the error is faster than the case without the control. On the other hand, since it is difficult to align the experimental conditions in the actual robot, it is inevitable that the performance in the actual robot is lower than that in the simulation, and there is room for improvement in the future.

The limitation of this study is described below. First, there is a problem that the speed of the iterative backpropagation becomes a rate-limiting factor and the balance control cannot be executed at a fast frequency. In this study, the limit is about 15 Hz, and the results are not much different from those of 5 Hz. It is found that if the period can be increased to about 100 Hz, the response to disturbances becomes faster, and the range of application will be expanded. On the other hand, the prediction accuracy of a trained model is likely an issue to be addressed in the future, since prediction errors

accumulate and a long control horizon is required for high frequency.

Second, there is a problem of data collection. In this study, we collected a variety of data by gradually shaking the body, but in order to obtain more dynamic data, we need to devise further ways of data collection, such as alternating between learning and data collection. If data collection becomes more efficient, it will be possible to handle not only simple balance control, but also more complex tasks such as stepping forward and walking, which require higher dimensional control inputs. In the future, it would be desirable to develop a curriculum learning method in which the robot learns to step while using a handrail, and gradually releases its hands when walking.

We describe some future developments. It would be meaningful to practice scenarios in which the robot wears different shoes depending on the task, such as shoes that are easy to balance, shoes that allow fast movement, waterproof shoes, and so on. In addition, we would like to consider the environment as a part of the body, and work on walking considering changes in the ground, using assitive tools, etc.

V. CONCLUSION

In this study, we proposed a deep predictive model learning method including parametric bias for balance control of complex musculoskeletal humanoids with flexibility and redundancy. For the task of balance control, it is difficult to collect data for in the actual robot. We can construct a stable balance control by collecting data while gradually increasing the random displacement of the control input and periodically changing its random width. In addition, the changes in upper body posture, the origin of joints and muscles, and footwear, which are not included in the dynamics model of the ankle, can be embedded as implicit changes in dynamics into parametric bias. Using the proposed DPMPB, the musculoskeletal humanoid successfully controls its balance according to various loss functions while adapting to changes in the body state. In the future, we would like to explore a method for autonomous learning of the foot-stepping motion using only the actual robot with assistance such as a handrail.

REFERENCES

[1] H. G. Marques, M. Jäntsh, S. Wittmeier, O. Holland, C. Alessandro, A. Diamond, M. Lungarella, and R. Knight, "ECCE1: the first of a series of anthropomimetic musculoskeletal upper torsos," in *Proceedings of the 2010 IEEE-RAS International Conference on Humanoid Robots*, 2010, pp. 391–396.

[2] Y. Asano, T. Kozuki, S. Ookubo, M. Kawamura, S. Nakashima, T. Katayama, Y. Iori, H. Toshinori, K. Kawaharazuka, S. Makino, Y. Kakiuchi, K. Okada, and M. Inaba, "Human Mimetic Musculoskeletal Humanoid Kengoro toward Real World Physically Interactive Actions," in *Proceedings of the 2016 IEEE-RAS International Conference on Humanoid Robots*, 2016, pp. 876–883.

[3] K. Kawaharazuka, S. Makino, K. Tsuzuki, M. Onitsuka, Y. Nagamatsu, K. Shinjo, T. Makabe, Y. Asano, K. Okada, K. Kawasaki, and M. Inaba, "Component Modularized Design of Musculoskeletal Humanoid Platform Musashi to Investigate Learning Control Systems," in *Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2019, pp. 7294–7301.

[4] Y. Nakanishi, I. Mizuuchi, T. Yoshikai, T. Inamura, and M. Inaba, "Pedaling by a redundant musculo-skeletal humanoid robot," in *Proceedings of the 2005 IEEE-RAS International Conference on Humanoid Robots*, 2005, pp. 68–73.

[5] A. Diamond and O. E. Holland, "Reaching control of a full-torso, modelled musculoskeletal robot using muscle synergies emergent under reinforcement learning," *Bioinspiration & Biomimetics*, vol. 9, no. 1, pp. 1–16, 2014.

[6] A. Marjaninejad, D. Urbina-Melëndez, B. A. Cohn, and F. J. Valero-Cuevas, "Autonomous functional movements in a tendon-driven limb via limited experience," *Nature Machine Intelligence*, vol. 1, no. 3, pp. 144–154, 2019.

[7] K. Kawaharazuka, S. Makino, M. Kawamura, Y. Asano, K. Okada, and M. Inaba, "Online Learning of Joint-Muscle Mapping using Vision in Tendon-driven Musculoskeletal Humanoids," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 772–779, 2018.

[8] K. Kawaharazuka, K. Tsuzuki, M. Onitsuka, Y. Asano, K. Okada, K. Kawasaki, and M. Inaba, "Musculoskeletal AutoEncoder: A Unified Online Acquisition Method of Intersensory Networks for State Estimation, Control, and Simulation of Musculoskeletal Humanoids," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2411–2418, 2020.

[9] K. Kawaharazuka, K. Tsuzuki, M. Onitsuka, Y. Asano, K. Okada, K. Kawasaki, and M. Inaba, "Object Recognition, Dynamic Contact Simulation, Detection, and Control of the Flexible Musculoskeletal Hand Using a Recurrent Neural Network With Parametric Bias," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4580–4587, 2020.

[10] Y. Asano, S. Nakashima, I. Yanokura, M. Onitsuka, K. Kawaharazuka, K. Tsuzuki, Y. Koga, Y. Omura, K. Okada, and M. Inaba, "Ankle-Hip-Stepping Stabilizer on Tendon-Driven Humanoid Kengoro by Integration of Muscle-Joint-Work Space Controllers for Knee-Stretched Humanoid Balance," in *Proceedings of the 2019 IEEE-RAS International Conference on Humanoid Robots*, 2019, pp. 397–402.

[11] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter, "Learning agile and dynamic motor skills for legged robots," *Science Robotics*, vol. 4, no. 26, 2019.

[12] OpenAI, I. Akkaya, M. Andrychowicz, M. Chociej, M. Litwin, B. McGrew, A. Petron, A. Paino, M. Plappert, G. Powell, R. Ribas, J. Schneider, N. Tezak, J. Tworek, P. Welinder, L. Weng, Q. Yuan, W. Zaremba, and L. Zhang, "Solving Rubik's Cube with a Robot Hand," arXiv preprint arXiv:1910.07113, 2019.

[13] J. Ahn, J. Lee, and L. Sentis, "Data-Efficient and Safe Learning for Humanoid Locomotion Aided by a Dynamic Balancing Model," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4376–4383, 2020.

[14] T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning robust perceptive locomotion for quadrupedal robots in the wild," *Science Robotics*, vol. 7, no. 62, 2022.

[15] Y. Yang, K. Caluwaerts, A. Iscen, T. Zhang, J. Tan, and V. Sindhwani, "Data Efficient Reinforcement Learning for Legged Robots," in *Proceedings of the 2019 Conference on Robot Learning*, 2019, pp. 1–10.

[16] J. Tani, "Self-organization of behavioral primitives as multiple attractor dynamics: a robot experiment," in *Proceedings of the 2002 International Joint Conference on Neural Networks*, 2002, pp. 489–494.

[17] K. Kawaharazuka, A. Miki, M. Bando, K. Okada, and M. Inaba, "Dynamic Cloth Manipulation Considering Variable Stiffness and Material Change Using Deep Predictive Model With Parametric Bias," *Frontiers in Neurorobotics*, vol. 16, pp. 1–16, 2022.

[18] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *Proceedings of the 3rd International Conference on Learning Representations*, 2015, pp. 1–15.

[19] K. Kawaharazuka, K. Okada, and M. Inaba, "Adaptive Robotic Tool-Tip Control Learning Considering Online Changes in Grasping State," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5992–5999, 2021.

[20] K. Kawaharazuka, K. Tsuzuki, S. Makino, M. Onitsuka, Y. Asano, K. Okada, K. Kawasaki, and M. Inaba, "Long-time Self-body Image Acquisition and its Application to the Control of Musculoskeletal Structures," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 2965–2972, 2019.

[21] E. Todorov, T. Erez, and Y. Tassa, "MuJoCo: A physics engine for model-based control," in *Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 5026–5033.

[22] M. Fiducioso, S. Curi, B. Schumacher, M. Gwerder, and A. Krause, "Safe Contextual Bayesian Optimization for Sustainable Room Temperature PID Control Tuning," in *Proceedings of 2019 International Joint Conference on Artificial Intelligence*, 2019, pp. 5850–5856.