Diffusion-Based Body Schema Learning Enabling Abnormal-State Adaptation in Musculoskeletal Robots

Kento Kawaharazuka[1] and Shuhei Ikemoto[2]

Abstract— **Musculoskeletal robots require an internal body schema that remains consistent under a wide range of physical state changes, including abnormalities such as muscle rupture and actuator jamming. Conventional approaches based on autoencoders or variational autoencoders learn average behaviors by projecting sensor and actuator signals into a low-dimensional latent space; however, exploration within the latent space alone has limited capability to handle out-of-distribution or abnormal states that are not included in the training data. To address this limitation, this study proposes a diffusion-based framework for body schema learning in musculoskeletal robots. Unlike generative models that operate through low-dimensional latent spaces, diffusion models can directly and iteratively estimate physically consistent sensor and actuator values in the high-dimensional space through a denoising process, even under partial observations and constraints, without requiring retraining. By formulating body schema adaptation as a gradient-guided denoising process, the proposed method enables adaptive estimation of appropriate muscle lengths and muscle tensions even under abnormal conditions such as muscle rupture and actuator jamming. The validity of the proposed framework is verified through simulation experiments using a musculoskeletal robot model.**

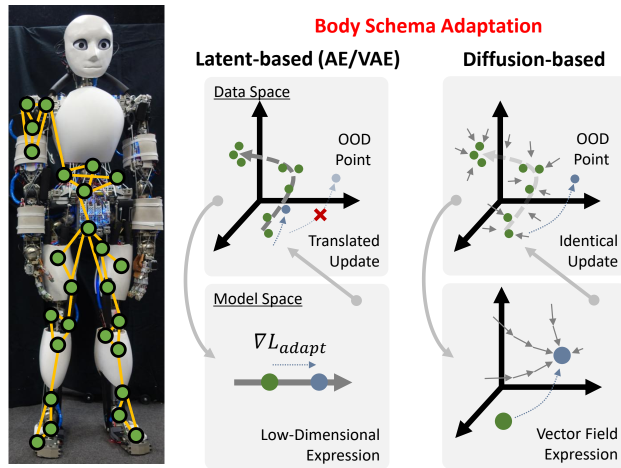

Fig. 1. Conceptual comparison of musculoskeletal body schema adaptation between low-dimensional latent-based generative models and high-dimensional diffusion-based models. Latent-based models perform updates in a compressed model space, restricting reachable solutions in the data space, whereas diffusion models operate directly in the high-dimensional data space, where all states remain accessible and are shaped through dynamical guidance.

I. INTRODUCTION

To date, a wide variety of musculoskeletal robots that imitate diverse human functions and control mechanisms have been proposed [1]–[4]. Because these robots are based on human anatomical models, they exhibit various characteristics such as muscle redundancy, nonlinear elasticity, and anisotropy [5]. This enables variable stiffness control at the hardware level [6], [7], as well as the direct implementation and experimental validation of human reflex structures [8]–[10]. Among these features, high adaptability arising from muscle redundancy is one of the greatest advantages of musculoskeletal robots. They are expected to continue operating in a human-like manner by adapting to abnormal conditions such as muscle rupture and actuator jamming [11].

To fully exploit this adaptability, it is necessary not only to handle redundancy at the hardware level but also to achieve adaptation at the software level, namely at the level of the body schema [12], [13]. Accordingly, a wide range of studies has been conducted. In general, a body schema realizes appropriate postures by learning the relationships among sensor and actuator variables such as joint angles, muscle lengths, and muscle tensions, and various representations have been employed, including data tables [14], polynomial approximations [15], and neural networks [16]. In particular, approaches using neural networks with high representational capacity have become common, and simple regression-based methods [17], autoencoder-based methods [18], and variational autoencoder-based methods [19] have been proposed. While simple regression-based methods are easy to train, adapting to muscle abnormalities requires either introducing learning variables that explicitly account for muscle rupture at the network input [20] or retraining the network using data that include abnormal states [11]. In contrast, autoencoder-based and variational autoencoder-based methods learn the redundant space of sensor and actuator signals as low-dimensional latent variables, and by manipulating these latent variables, they may adapt to abnormal states without retraining. However, because these methods learn average behavior by projecting the redundant space into a low-dimensional latent space, their ability to handle out-of-distribution abnormal states that are not included in the training data is likely to be limited, as illustrated in the left panel of Fig. 1.

Accordingly, this study proposes a diffusion-based framework for body schema learning in musculoskeletal robots that enables adaptation to abnormal states. Unlike generative models that rely on low-dimensional latent spaces, such as autoencoders and variational autoencoders, diffusion models

[1] The author is 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@jsk.imi.i.u-tokyo.ac.jp

[2] The author is with the Graduate School of Life Science and Systems Engineering, Kyushu Institute of Technology, 7230, Hibikino 2-4, Wakamatsu, Kitakyushu, Fukuoka, 808-0135, Japan. ikemoto@brain.kyutech.ac.jp

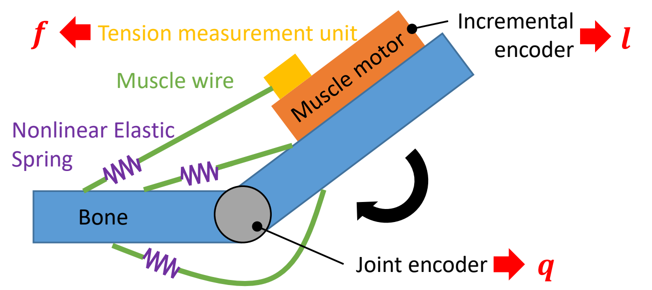

Fig. 2. The basic structure of musculoskeletal robots used in this study. Muscles are modeled as wires that connect motors to the skeleton via a muscle tension measurement unit and a nonlinear elastic spring at the tip. Muscle length is measured from the motor encoder, and muscle tension is measured from the pressure sensor attached to the muscle tension measurement unit.

iteratively estimate physically consistent sensor and actuator values directly in the original high-dimensional space where they are defined, through a denoising process. By formulating body schema estimation as a gradient-guided denoising process, we hypothesize that appropriate muscle lengths and muscle tensions can be adaptively estimated even under abnormal conditions such as muscle rupture and actuator jamming. Through simulations of a two-joint, six-muscle musculoskeletal robot model, we demonstrate that the proposed diffusion-based body schema can adapt its internal representation and maintain consistent behavior across various actuator failure scenarios without the need for explicit failure annotations during pretraining or any subsequent retraining. Furthermore, we validate the effectiveness of the proposed method through comparisons with body schema adaptation approaches based on autoencoders and variational autoencoders.

This paper is organized as follows. Section II describes the basic structure of musculoskeletal robots and body schema learning methods based on autoencoders, variational autoencoders, and diffusion models. Section III presents latent-based and gradient-based body schema adaptation methods for abnormal conditions such as muscle rupture and actuator jamming. Section IV details the musculoskeletal robot simulation, body schema learning settings, and evaluation methodologies. Section V reports comparative experiments of the proposed and baseline body schema learning methods using musculoskeletal robot simulations, followed by discussion.

II. Body Schema Learning based on AutoEncoder, Variational AutoEncoder, and Diffusion Model

A. Basic Structure and Body Schema of Musculoskeletal Robots

We first describe the assumptions of this study. As shown in Fig. 2, a musculoskeletal structure is basically composed of a skeleton, joints, and wires that emulate muscles. Depending on the robot, joint angles $\boldsymbol{q}$ may be directly available; even when they are not, they can be estimated by combining muscle length changes with visual sensors [16]. Each wire is wound around a pulley attached to the motor shaft and is connected to the skeleton through a muscle tension measurement unit. In this configuration, the muscle length l^{motor} is obtained from an encoder attached to the motor, while the muscle tension f is obtained from a pressure sensor attached to the muscle tension measurement unit. A nonlinear elastic spring is attached to the end of each wire, and this, together with redundancy, enables variable stiffness control. The relationship between muscle tension and the extension of the nonlinear elastic spring is modeled as follows:

$$f_i = k(\exp(\alpha\delta_i) - 1) \tag{1}$$

Here, k denotes the elastic coefficient, α is a constant representing the degree of nonlinearity, and δ represents the spring extension.

Next, let the number of joints be N and the number of muscles be M, and consider the relationship among the joint angles $\boldsymbol{q} \in \mathbb{R}^N$, the muscle lengths $\boldsymbol{l}^{motor} \in \mathbb{R}^M$, and the muscle tensions $\boldsymbol{f} \in \mathbb{R}^M$. For muscle i, the following holds:

$$\delta_i = \phi^{-1}(f_i) = \max(0, l_i^{geo}(\boldsymbol{q}) - l_i^{motor}) \tag{2}$$

Here, $\boldsymbol{l}^{geo}(\boldsymbol{q})$ denotes a function representing the geometric muscle lengths that depend on the joint angles. When nonlinear elasticity and wire elongation are not considered, the muscle lengths $\boldsymbol{l}^{motor}$ obtained from the motors are equal to the geometric muscle lengths. In addition, when the muscle tension is positive, the inverse mapping can be computed as follows:

$$\delta_i = \frac{1}{\alpha}\log(\frac{f_i}{k} + 1) \tag{3}$$

Finally, we explain the objective of body schema learning in this study. The goal of body schema learning in musculoskeletal robots is to learn the relationships among the joint angles $\boldsymbol{q}$, muscle lengths $\boldsymbol{l}^{motor}$, and muscle tensions $\boldsymbol{f}$, so that these quantities can be appropriately estimated. In particular, this study aims to estimate the muscle lengths $\boldsymbol{l}^{motor}$ and muscle tensions $\boldsymbol{f}$ of the robot that realize a given target joint angle $\boldsymbol{q}^{ref}$. That is, we learn a body schema that takes $\boldsymbol{q}$ as input and outputs $\boldsymbol{l}^{motor}$ and $\boldsymbol{f}$. Although evaluation in this study is conducted using the model described in Eq. (1)–Eq. (3), none of this information is available during training or during adaptation to abnormal conditions. Using only the learned body schema that captures the relationships among $\boldsymbol{q}$, $\boldsymbol{l}^{motor}$, and $\boldsymbol{f}$, we estimate $\boldsymbol{l}^{motor}$ and $\boldsymbol{f}$ that achieve $\boldsymbol{q}^{ref}$ while adapting to the given abnormal state, regardless of whether muscle length control or muscle tension control is employed.

B. Body Schema Learning based on AutoEncoder and Variational AutoEncoder

First, we describe body schema learning methods based on autoencoders (AE) and variational autoencoders (VAE), as shown in the left and center panels of Fig. 3. The

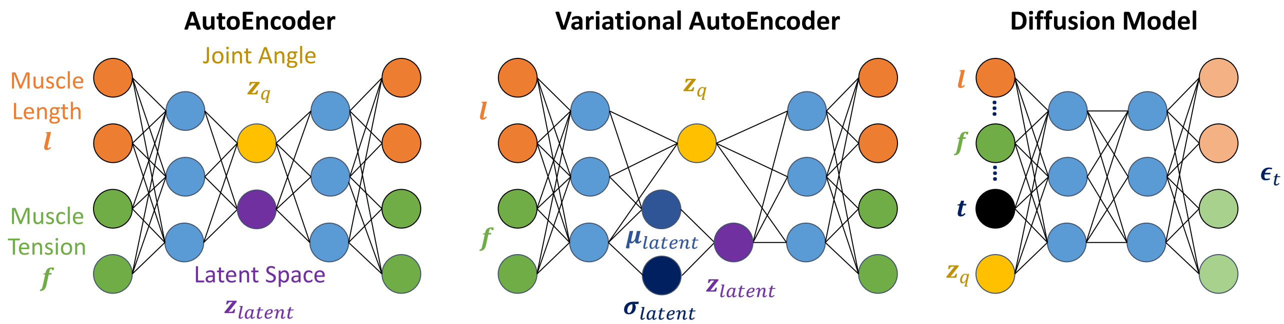

Fig. 3. Body schema learning methods based on autoencoders (left), variational autoencoders (center), and diffusion models (right).

autoencoder-based method is constructed as follows:

$$\boldsymbol{x} = [\boldsymbol{l}^{motor}, \boldsymbol{f}] \tag{4}$$

$$\boldsymbol{z}_q, \boldsymbol{z}_{latent} = \boldsymbol{h}^{ae}_{enc}(\boldsymbol{x}) \tag{5}$$

$$\hat{\boldsymbol{x}} = \boldsymbol{h}^{ae}_{dec}(\boldsymbol{z}_q, \boldsymbol{z}_{latent}) \tag{6}$$

Here, $\boldsymbol{h}^{ae}_{enc}$ and $\boldsymbol{h}^{ae}_{dec}$ denote the encoder and decoder, respectively. The latent variable $\boldsymbol{z}_q$ corresponds to the joint angles $\boldsymbol{q}$, while $\boldsymbol{z}_{latent}$ represents the latent variables corresponding to the redundant space. Note that $\boldsymbol{x}$ is normalized using the training data, and hereafter $\boldsymbol{x}$ is assumed to be normalized during all training procedures.

During training, the following loss function is minimized:

$$L^{ae}_{train} = ||\boldsymbol{x} - \hat{\boldsymbol{x}}||^2 + C^{train}_q ||\boldsymbol{q} - \boldsymbol{z}_q||^2 \tag{7}$$

Here, C^{train}_q is a constant that represents the weight for joint angle estimation. By training this autoencoder, muscle lengths $\boldsymbol{l}^{motor}$ and muscle tensions $\boldsymbol{f}$ can be computed by providing the joint angles $\boldsymbol{q}$ and the latent variables $\boldsymbol{z}_{latent}$ corresponding to the redundant space as inputs to $\boldsymbol{h}^{ae}_{dec}$.

Next, the variational autoencoder-based method is constructed as follows:

$$\boldsymbol{z}_q, \boldsymbol{\mu}_{latent}, \boldsymbol{\sigma}_{latent} = \boldsymbol{h}^{vae}_{enc}(\boldsymbol{x}) \tag{8}$$

$$\boldsymbol{z}_{latent} \sim \mathcal{N}(\boldsymbol{\mu}_{latent}, \mathrm{diag}(\boldsymbol{\sigma}^2_{latent})) \tag{9}$$

$$\hat{\boldsymbol{x}} = \boldsymbol{h}^{vae}_{dec}(\boldsymbol{z}_q, \boldsymbol{z}_{latent}) \tag{10}$$

Here, $\boldsymbol{h}^{vae}_{enc}$ and $\boldsymbol{h}^{vae}_{dec}$ denote the encoder and decoder, respectively.

During training, the following loss function is minimized:

$$\begin{aligned} L^{vae}_{train} = & ||\boldsymbol{x} - \hat{\boldsymbol{x}}||^2 + C^{train}_q ||\boldsymbol{q} - \boldsymbol{z}_q||^2 \\ & + D_{KL}(\mathcal{N}(\boldsymbol{\mu}_{latent}, \mathrm{diag}(\boldsymbol{\sigma}^2_{latent}))||\mathcal{N}(\boldsymbol{0}, I)) \end{aligned} \tag{11}$$

Here, D_{KL} denotes the Kullback-Leibler divergence. Similar to the autoencoder-based method, muscle lengths $\boldsymbol{l}^{motor}$ and muscle tensions $\boldsymbol{f}$ can be computed by providing the joint angles $\boldsymbol{q}$ and the latent variables $\boldsymbol{z}_{latent}$ corresponding to the redundant space as inputs to $\boldsymbol{h}^{vae}_{dec}$.

C. Body Schema Learning based on Diffusion Model

We describe the body schema learning method based on diffusion models, as shown in the right panel of Fig. 3. A diffusion model consists of a noising process and a denoising process. First, the noising process is defined as follows:

$$\boldsymbol{x}_t = \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, I) \tag{12}$$

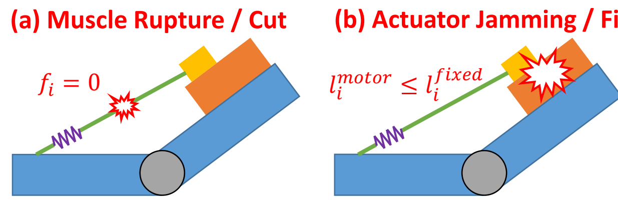

Fig. 4. Abnormal states considered in this study: (a) muscle rupture (Cut) and (b) actuator jamming (Fix).

Here, $\boldsymbol{x}_0 = [\boldsymbol{l}^{motor}, \boldsymbol{f}]$ represents the sensor and actuator values normalized using the training data, and $\boldsymbol{x}_t$ denotes the sensor and actuator values after noise addition at time step t. The constant $\bar{\alpha}_t$ is determined by a scheduler.

Next, the denoising process is defined as follows:

$$\hat{\boldsymbol{\epsilon}}_t = \boldsymbol{h}^{diff}(\boldsymbol{x}_t, t, \boldsymbol{q}) \tag{13}$$

$$\boldsymbol{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\hat{\boldsymbol{\epsilon}}_t\right) + \sigma_t\boldsymbol{\eta}, \quad \boldsymbol{\eta} \sim \mathcal{N}(\boldsymbol{0}, I) \tag{14}$$

Here, $\boldsymbol{h}^{diff}$ denotes the diffusion model, and α_t and σ_t are constants determined by the scheduler. By training this diffusion model, the noise term $\hat{\boldsymbol{\epsilon}}_t$ can be estimated given the joint angles $\boldsymbol{q}$ and the noise-corrupted sensor and actuator values $\boldsymbol{x}_t$ as inputs. By iteratively applying the denoising process, the initial state $\boldsymbol{x}_0$ corresponding to $\boldsymbol{l}^{motor}$ and $\boldsymbol{f}$ can be estimated.

III. Body Schema Adaptation to Abnormal States

Here, we describe body schema adaptation methods for coping with abnormal conditions such as muscle rupture and actuator jamming. In this study, rather than retraining the model using newly collected data under abnormal conditions, we aim to adapt to such conditions by utilizing a pretrained body schema. This approach is important because it enables immediate adaptation when abnormal conditions occur, because appropriate training data for abnormal states may not be available in the first place, and because it helps avoid overfitting caused by retraining.

A. Definition of Abnormal States

We describe the abnormal conditions considered in this study, as illustrated in Fig. 4. In this study, two types of abnormal conditions are addressed: muscle rupture and actuator jamming.

First, muscle rupture refers to a state in which a muscle is completely severed and the muscle tension becomes zero. When muscle i is ruptured, the following condition holds:

$$f_i = 0 \tag{15}$$

Next, actuator jamming refers to a state in which the motor becomes immobile and the muscle length is fixed. When muscle i is jammed, the following condition holds:

$$l_i^{motor} \leq l_i^{fixed} \tag{16}$$

Here, l_i^{fixed} is a constant representing the muscle length at the moment of jamming. This constraint is formulated as an inequality constraint because, due to tension anisotropy of the wire, the muscle length can be shortened but cannot be extended.

B. Body Schema Adaptation based on Latent Space Optimization

We describe body schema adaptation methods based on autoencoders and variational autoencoders for adapting to the abnormal conditions defined in Section III-A. Adaptation of these body schemas is performed via latent space optimization. Specifically, the latent variable $\boldsymbol{z}_{latent}$ is updated as follows.

$$\begin{aligned}
\hat{\boldsymbol{x}} =& \boldsymbol{h}_{dec}^{\{ae,vae\}}(\boldsymbol{q}, \boldsymbol{z}_{latent}) \qquad (17)\\
L_{adapt}(\hat{\boldsymbol{x}}) =& C_{cut}^{adapt} \sum_{i \in \mathcal{I}_{cut}} ||f_i||^2 \\
&+ C_{fix}^{adapt} \sum_{j \in \mathcal{I}_{fix}} ||\max(0, l_j^{motor} - l_j^{fixed})||^2 \\
&+ C_{fmin}^{adapt} \sum_{i=1}^{M} \max(0, -f_i)^2 \\
&+ C_{fmax}^{adapt} \sum_{i=1}^{M} \max(0, f_i - f_i^{max})^2 \\
\boldsymbol{z}_{latent} \leftarrow& \boldsymbol{z}_{latent} - C_{lr}^{adapt} \nabla_{\boldsymbol{z}_{latent}} L_{adapt}(\hat{\boldsymbol{x}}) \qquad (18)
\end{aligned}$$

Here, $\mathcal{I}_{cut}$ denotes the index set of ruptured muscles, $\mathcal{I}_{fix}$ denotes the index set of jammed muscles, and f^{max} is a constant representing the maximum muscle tension. In addition, C_{cut}^{adapt}, C_{fix}^{adapt}, C_{fmin}^{adapt}, and C_{fmax}^{adapt} are constants representing the weights for each evaluation term, and C_{lr}^{adapt} is a constant representing the learning rate. In other words, adaptation to abnormal conditions is achieved by updating the latent variable $\boldsymbol{z}_{latent}$ so as to satisfy the constraints of muscle rupture and actuator jamming, while ensuring that the muscle tension remains nonnegative and does not exceed the maximum muscle tension.

C. Body Schema Adaptation based on Gradient-Guided Denoising Process

We describe the body schema adaptation method based on diffusion models for adapting to the abnormal conditions defined in Section III-A. Diffusion-based body schema adaptation is performed by combining gradient-based guidance with the denoising process. Specifically, the denoising process is modified as follows:

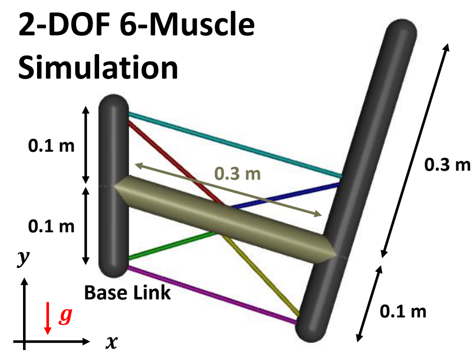

Fig. 5. Experimental setup using a two-joint, six-muscle musculoskeletal model.

$$\hat{\boldsymbol{x}}_0 = \frac{\boldsymbol{x}_t - \sqrt{1-\bar{\alpha}_t}\hat{\boldsymbol{\epsilon}}}{\sqrt{\bar{\alpha}_t}} \tag{19}$$

$$\boldsymbol{x}_t \leftarrow \boldsymbol{x}_t - C_{lr}^{adapt} \nabla_{\boldsymbol{x}_t} L_{adapt}(\hat{\boldsymbol{x}}_0) \tag{20}$$

$$\boldsymbol{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\hat{\boldsymbol{\epsilon}}_t\right) \tag{21}$$

Here, $L_{adapt}(\hat{\boldsymbol{x}}_0)$ uses the loss function defined in Eq. (18). In this process, $\boldsymbol{\eta}$ is set to zero, and no additional noise is injected.

IV. Experimental Setup

The simulation environment using the two-joint, six-muscle model considered in this study is shown in Fig. 5. From a base link of length 0.2 m, two links of lengths 0.3 m and 0.4 m, each with a mass of 2.0 kg, are serially connected by rotational joints, and four monoarticular muscles and two biarticular muscles are attached. Each joint is movable within the range of [-1.0, 1.0] rad, and gravity acts in the negative direction of the y-axis. The parameters are set to $f^{max} = 150$ N, $k = 1000$ N, and $\alpha = 20.0$ m^{-1}.

A. Data Collection and Training of Body Schema

We describe the data collection procedure for body schema learning. In data collection, N^{data} samples of $(\boldsymbol{q}, \boldsymbol{f}, \boldsymbol{l}^{motor})$ are generated. The joint angles $\boldsymbol{q}$ are uniformly sampled, and the geometric muscle lengths $\boldsymbol{l}^{geo}(\boldsymbol{q})$ and their derivatives, namely the muscle Jacobian $\boldsymbol{G}(\boldsymbol{q})$, are obtained using forward kinematics in MuJoCo. Then, muscle tensions $\boldsymbol{f}$ that satisfy static equilibrium are computed by solving the following optimization problem:

$$\begin{aligned}
\arg\min_{\boldsymbol{f} \in \mathbb{R}^M} \quad & ||\boldsymbol{f} - \boldsymbol{f}^{ref}||_2^2 \qquad (22)\\
\text{s.t.} \quad & -\boldsymbol{G}(\boldsymbol{q})^\top \boldsymbol{f} = \boldsymbol{\tau}^{bias}(\boldsymbol{q}) \\
& \boldsymbol{0} \leq \boldsymbol{f} \leq \boldsymbol{f}^{\max}
\end{aligned}$$

Here, $\boldsymbol{f}^{ref}$ denotes a random vector sampled from the range $[\boldsymbol{0}, \boldsymbol{f}^{max}]$, and $\boldsymbol{\tau}^{bias}(\boldsymbol{q})$ represents the torque that the robot should generate, which in this study corresponds to the gravity compensation torque. Finally, the muscle lengths $\boldsymbol{l}^{motor}$ are computed from Eq. (2). In this way, a dataset with diverse muscle tension patterns that satisfy the required torques is generated and used for training.

Next, we describe the model architectures of the three body schema models: autoencoder, variational autoencoder, and diffusion model. The autoencoder and variational autoencoder consist of encoders and decoders implemented as three-layer multilayer perceptrons (MLPs), where each hidden layer has 64 units and the activation function is Tanh. For these models, comparisons are conducted with latent space dimensions of 3, 10, and 30, resulting in approximately 10k, 11k, and 14k total parameters, respectively, and approximately 11k, 12k, and 15k parameters in the case of the variational autoencoder. Note that the latent space dimensionality is the sum of the dimensions of z_q and z_{latent}. Since the dimensionality of z_q is 2 in this study, the dimensionality of z_{latent} becomes 1, 8, and 28, respectively. The diffusion model is implemented as a three-layer MLP with two hidden layers of 91 units each, uses the SiLU activation function, and has approximately 11k parameters. Across all models, the number of parameters is kept as similar as possible.

These three models are trained using $N^{data} = 800$, with 80 percent of the data used for training and 20 percent for evaluation, over 10000 epochs. For the diffusion model, the number of diffusion steps is set to 200, and a linear scheduler is used. Other parameters are set to $C_q^{train} = 2.0$, $C_{cut}^{adapt} = 0.01$, $C_{fix}^{adapt} = 1.0$, $C_{fmin}^{adapt} = 0.01$, $C_{fmax}^{adapt} = 0.01$, and $C_{lr}^{adapt} = 0.01$.

B. Evaluation

We describe the evaluation methodology for body schema learning. Here, we evaluate whether the muscle lengths $\boldsymbol{l}^{motor}$ and muscle tensions $\boldsymbol{f}$ computed in Section III can satisfy the conditions in Eq. (15) and Eq. (16), achieve joint torque equilibrium, and realize the target joint angles $\boldsymbol{q}^{ref}$. An abnormal condition in which two randomly selected muscles out of six are ruptured is defined as **CUT**, an abnormal condition in which two muscles are jammed is defined as **FIX**, and an abnormal condition in which one muscle is ruptured and one muscle is jammed is defined as **BOTH**. Adaptation is performed for each abnormal condition. The adaptation performance of each model is evaluated using 40 sampled abnormal cases.

For evaluation, the joint angles $\hat{\boldsymbol{q}}$ that would be realized given $\boldsymbol{l}^{motor}$ and $\boldsymbol{f}$ are computed using forward kinematics in MuJoCo, static equilibrium, and the abnormal condition constraints. This is achieved by iteratively updating $\hat{\boldsymbol{q}}$ using the Gauss-Newton method under multiple constraints. Since physical simulation including muscle slack is difficult, an optimization-based approach is adopted. First, forward kinematics in MuJoCo is used to obtain the geometric muscle lengths $\boldsymbol{l}^{geo}(\hat{\boldsymbol{q}})$, the muscle Jacobian $\boldsymbol{G}(\hat{\boldsymbol{q}})$, and bias torques such as gravity compensation torque $\boldsymbol{\tau}^{bias}(\hat{\boldsymbol{q}})$. Here, the set of taut muscles $\mathcal{E}(\hat{\boldsymbol{q}})$ is defined based on a tension threshold and geometric conditions as follows:

$$\mathcal{E}(\hat{\boldsymbol{q}}) = \Big\{ i \notin (\mathcal{I}_{cut} \cup \mathcal{I}_{fix}) \;\Big|\; f_i > 0, \\ l_i^{geo}(\hat{\boldsymbol{q}}) - l_i^{motor} \geq -\varepsilon^{slack} \Big\} \tag{23}$$

Here, ε^{slack} is a constant representing the slack tolerance. For taut muscles $i \in \mathcal{E}(\hat{\boldsymbol{q}})$, equality residuals are defined using the elongation $\delta_i = \phi^{-1}(f_i)$:

$$r_i^{eq}(\hat{\boldsymbol{q}}) = \big(l_i^{motor} + \phi^{-1}(f_i)\big) - l_i^{geo}(\hat{\boldsymbol{q}}), \\ i \in \mathcal{E}(\hat{\boldsymbol{q}}) \tag{24}$$

For slack muscles $i \notin \mathcal{E}(\hat{\boldsymbol{q}})$, the violation of the slack condition is introduced as a hinge residual:

$$r_i^{slack}(\hat{\boldsymbol{q}}) = \max\big(0,\ l_i^{geo}(\hat{\boldsymbol{q}}) - l_i^{motor}\big), \\ i \notin \mathcal{E}(\hat{\boldsymbol{q}}), i \notin \mathcal{I}_{cut} \cup \mathcal{I}_{fix} \tag{25}$$

For jammed muscles $i \in \mathcal{I}_{fix}$, a one-sided constraint is imposed using the fixed motor length l_i^{fix}:

$$r_i^{fix}(\hat{\boldsymbol{q}}) = \max\Big(0,\ l_i^{geo}(\hat{\boldsymbol{q}}) - \Big(l_i^{fix} + \phi^{-1}(f_i)\Big)\Big), \\ i \in \mathcal{I}_{fix} \tag{26}$$

These residuals are solved as an Iteratively Reweighted Least Squares (IRLS) problem with weights w_i assigned to each residual, minimizing the Huber loss [21].

In addition, a static equilibrium regularization term is introduced. We define the effective muscle tension $\boldsymbol{f}^{eff}$ using only taut muscles as

$$f_i^{eff} = \begin{cases} f_i & (i \in \mathcal{E}(\hat{\boldsymbol{q}})) \\ 0 & \text{otherwise} \end{cases} \tag{27}$$

Using this definition, the joint torque imbalance between the muscle-generated torque and the bias torque is given by

$$\boldsymbol{\tau}(\hat{\boldsymbol{q}}) = -\boldsymbol{G}(\hat{\boldsymbol{q}})^\top \boldsymbol{f}^{eff} - \boldsymbol{\tau}^{bias}(\hat{\boldsymbol{q}}) \tag{28}$$

and is added as a regularization term.

To compensate for differences in residual units, a length scale s_l and a torque scale s_τ are introduced. Finally, $\hat{\boldsymbol{q}}$ is estimated by solving the following minimization problem:

$$\min_{\hat{\boldsymbol{q}} \in \mathbb{R}^2} \sum_i w_i \left(\frac{r_i(\hat{\boldsymbol{q}})}{s_l}\right)^2 + \lambda_\tau \left|\left|\frac{\boldsymbol{\tau}(\hat{\boldsymbol{q}})}{s_\tau}\right|\right|_2^2 + \lambda_q ||\hat{\boldsymbol{q}} - \boldsymbol{q}^{ref}||_2^2 \tag{29}$$

Here, $r_i(\hat{\boldsymbol{q}})$ is chosen from $\{r_i^{eq}, r_i^{slack}, r_i^{fix}\}$ depending on the muscle state, and ruptured muscles $i \in \mathcal{I}_{cut}$ are completely excluded. The constant λ_τ represents the weight of the static equilibrium regularization, and λ_q represents the weight of the joint angle regularization. The residual vector $\boldsymbol{r}(\hat{\boldsymbol{q}})$ is linearized as follows, and $\hat{\boldsymbol{q}}$ is iteratively updated using the Gauss-Newton method:

$$\boldsymbol{r}(\hat{\boldsymbol{q}} + \Delta\hat{\boldsymbol{q}}) \approx \boldsymbol{r}(\hat{\boldsymbol{q}}) + \boldsymbol{J}(\hat{\boldsymbol{q}})\Delta\hat{\boldsymbol{q}} \tag{30}$$

$$\big(\boldsymbol{J}^\top \boldsymbol{J} + \mu \boldsymbol{I}\big)\,\Delta\hat{\boldsymbol{q}} = -\boldsymbol{J}^\top \boldsymbol{r} \tag{31}$$

$$\hat{\boldsymbol{q}} \leftarrow \hat{\boldsymbol{q}} + \Delta\hat{\boldsymbol{q}} \tag{32}$$

Through this estimation, given the input $(\boldsymbol{l}^{motor}, \boldsymbol{f})$, the joint angles $\hat{\boldsymbol{q}}$ that simultaneously satisfy rupture and jamming conditions, slack constraints, and static equilibrium are reconstructed.

Next, we define four evaluation metrics, which represent the tension error under muscle rupture, the muscle length error under actuator jamming, the static equilibrium error, and the joint angle error, respectively.

$$E_{cut} = \frac{1}{|\mathcal{I}_{cut}|} \sum_{i \in \mathcal{I}_{cut}} |f_i| \tag{33}$$

$$E_{fix} = \frac{1}{|\mathcal{I}_{fix}|} \sum_{j \in \mathcal{I}_{fix}} \max(0, l_j^{motor} - l_j^{fixed}) \tag{34}$$

$$E_\tau = || - \boldsymbol{G}(\hat{\boldsymbol{q}})^\top \boldsymbol{f}^{eff} - \boldsymbol{\tau}^{bias}(\hat{\boldsymbol{q}})||_2 \tag{35}$$

$$E_q = ||\boldsymbol{q}^{ref} - \hat{\boldsymbol{q}}||_2 \tag{36}$$

In this study, the parameters are set to $\varepsilon^{slack} = 10^{-5}$ m, $s_l = 0.03$ m, $s_\tau = 5.0$ Nm, $\lambda_\tau = 0.1$, and $\lambda_q = 10^{-3}$.

V. Experimental Results and Discussion

A. Experimental Results

We compared the adaptation performance of each model $\{$autoencoder (AE), variational autoencoder (VAE), diffusion model (Diff)$\}$ under each abnormal condition $\{$**CUT**, **FIX**, **BOTH**$\}$. The latent space dimensionalities of the AE and VAE are compared for values of 3, 10, and 30, and the results are reported as AE-3, AE-10, AE-30, VAE-3, VAE-10, VAE-30, and Diff, respectively. The experimental results are shown in Fig. 6. For each evaluation metric E_{cut}, E_{fix}, E_τ, and E_q, adaptation results over 40 abnormal condition samples are presented using box-and-whisker plots. Also, Fig. 7 compares the inference times of AE-10, VAE-10, and Diff under the **BOTH** condition and under a condition without any adaptation. When no adaptation is performed, inference in AE and VAE consists only of Eq. (17), whereas inference in Diff consists solely of the iterative application of Eq. (19) and Eq. (21). Note that all inference times were measured on an Intel Core i7-10750H CPU and an NVIDIA Quadro T2000 Max-Q GPU.

First, for the tension error under muscle rupture E_{cut}, both under **CUT** and **BOTH** conditions, AE-10, AE-30, and Diff exhibit high adaptation performance. Through the latent space optimization or the denoising process, the muscle tension of ruptured muscles is successfully driven close to zero. In contrast, AE-3 and VAE-based methods in general fail to sufficiently reduce muscle tension, resulting in inferior adaptation performance.

Next, for the muscle length error under actuator jamming E_{fix}, under both **FIX** and **BOTH** conditions, all models except AE-10 and AE-30 show high adaptation performance. Unlike the case of E_{cut}, VAE-based methods exhibit better adaptation performance than AE-based methods.

Next, for the static equilibrium error E_τ, under all abnormal conditions **CUT**, **FIX**, and **BOTH**, AE-3, VAE-3, VAE-10, and Diff demonstrate high adaptation performance. In particular, Diff achieves superior adaptation performance.

Next, for the joint angle error E_q, under all abnormal conditions **CUT**, **FIX**, and **BOTH**, AE-3, VAE-3, VAE-10, and Diff show high adaptation performance. In particular, VAE-10 and Diff achieve the best adaptation performance.

Finally, regarding inference time, when no adaptation is performed, AE and VAE complete inference within 0.001 s, whereas Diff requires approximately 0.1 s. In contrast, when adaptation is enabled, the inference times of AE, VAE, and Diff are all approximately 0.7 s.

B. Discussion

Based on the results presented above, we discuss the adaptation performance of each model. For the tension error under muscle rupture E_{cut}, AEs with larger latent spaces and the diffusion model exhibited high adaptation performance. For the muscle length error under actuator jamming E_{fix}, VAE-based methods and the diffusion model showed high adaptation performance. For the static equilibrium error E_τ and the joint angle error E_q, AEs and VAEs with smaller latent spaces, as well as the diffusion model, demonstrated high adaptation performance. From these results, it can be observed that the diffusion model consistently achieves stable and strong adaptation performance across all evaluation metrics. It successfully satisfies the constraints imposed by muscle rupture and actuator jamming, static equilibrium conditions, and the target joint angles.

In contrast, AE-based methods fail to sufficiently satisfy either the muscle rupture or actuator jamming constraints, and their adaptation performance in terms of static equilibrium and joint angle accuracy is also limited. VAE-based methods show high adaptation performance for actuator jamming, but tend to perform poorly under muscle rupture conditions. For static equilibrium and joint angle errors, VAEs with a latent space dimensionality of 10 exhibit better adaptation performance than those with latent dimensions of 3 or 30. This suggests that selecting an appropriate latent space dimensionality is critical for achieving high adaptation performance with VAEs.

These results can be explained by several factors. To satisfy the constraints imposed by muscle rupture and actuator jamming, the body schema must be capable of representing these constraints. Since the latent space of AE-based methods is learned without explicit regularization, optimization can drive the latent variables to values that deviate significantly from the training distribution. This often leads to a dilemma in which satisfying the muscle rupture constraint prevents satisfying the actuator jamming constraint, or vice versa. In contrast, VAEs impose regularization on the latent space, and increasing the latent dimensionality improves representational capacity, making it easier to satisfy actuator jamming constraints. However, when the latent space becomes excessively large, adaptation performance for static equilibrium and joint angle accuracy deteriorates, indicating that careful selection of the latent dimensionality is required. Aggregating information into a Gaussian-distributed low-dimensional latent space is therefore likely to hinder stable adaptation to diverse abnormal conditions.

By contrast, diffusion models can represent diverse data distributions through the denoising process without relying on the size of a latent space. As a result, diffusion models can satisfy muscle rupture and actuator jamming constraints while simultaneously achieving high adaptation performance in static equilibrium and joint angle accuracy. Rather than

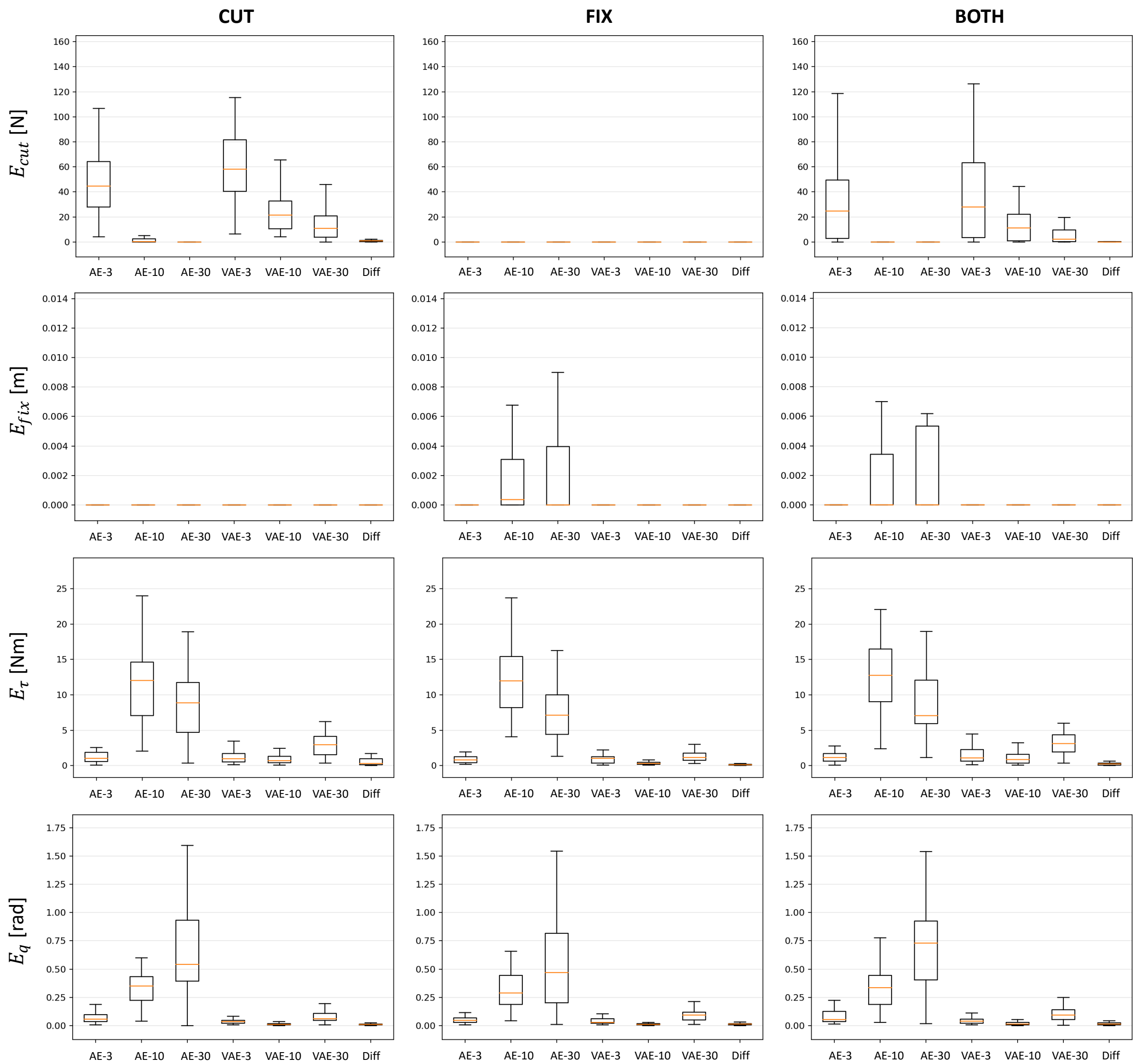

Fig. 6. Experimental results of adaptation performance under each abnormal condition. The box-and-whisker plots show the distribution of adaptation results over 40 abnormal condition samples for each evaluation metric E_{cut}, E_{fix}, E_τ, and E_q.

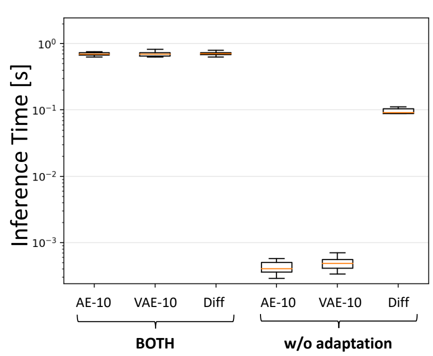

Fig. 7. Inference time comparison with and without adaptation.

compressing information into a latent space, directly utilizing gradient information during the denoising process contributes to stable adaptation under special and abnormal conditions. These results indicate that directly operating information in the high-dimensional space, without collapsing or distorting the original data, is effective for robots with redundant sensors and actuators.

Finally, we discuss the limitations and future challenges. This study is limited to simulation experiments. The primary reason is that reproducing a wide variety of abnormal conditions on real hardware is difficult, making quantitative evaluation challenging. Note that, in musculoskeletal systems, joints and muscles are typically grouped as in [22], with each group usually consisting of no more than approximately five joints and ten muscles. Since this is not substantially larger than the two-joint, six-muscle system considered in this study, the network complexity is not expected to differ significantly between simulation and real-robot applications. In addition, while this study focuses on muscle rupture and actuator jamming, real robotic systems may experience circuit failures, sensor noise, and various other malfunctions. In future work, it will be necessary to investigate which special conditions are likely to occur in practice and to demonstrate the proposed approach in a more practical setting. Furthermore, the proposed implementation requires approximately 0.7 s for adaptation, which is acceptable for static motions but insufficient for dynamic behaviors. Reducing the inference time will therefore be necessary, and recent advances in diffusion model acceleration may help address this limitation. Although this study focuses on musculoskeletal structures, the proposed methodology can be generally applied to robots with redundant sensors and actuators. Robots with redundant joints, redundant muscles, or redundant sensors represent a broad range of potential applications. We expect that the proposed method will further unlock the adaptive capabilities enabled by redundancy.

VI. CONCLUSION

In this study, we proposed methods for body schema learning and adaptation to abnormal states in musculoskeletal robots. To address the high dimensionality of the sensor-actuator space in musculoskeletal robots, we investigated body schema learning approaches based on autoencoders, variational autoencoders, and diffusion models. Furthermore, for abnormal conditions such as muscle rupture and actuator jamming, we proposed body schema adaptation methods based on latent-space optimization and gradient-guided denoising processes. These methods were applied to a simulated musculoskeletal robot model, revealing that autoencoders exhibit reduced adaptation capability as the latent space dimensionality increases, that variational autoencoders achieve good adaptation performance when an appropriate latent dimensionality is chosen, and that diffusion models consistently provide robust adaptation performance across all conditions. These results indicate that, for robots with redundant sensors and actuators, directly handling their relationships as high-dimensional information, rather than projecting them into a low dimensional latent space, enables gradient-based adaptation to a wide range of situations. In future work, we will continue to evaluate the proposed approach on a broader range of robot morphologies and abnormal conditions.

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 *2010 IEEE-RAS International Conference on Humanoid Robots*, 2010, pp. 391–396.

[2] M. Jäntsch, S. Wittmeier, K. Dalamagkidis, A. Panos, F. Volkart, and A. Knoll, "Anthrob - A Printed Anthropomimetic Robot," in *2013 IEEE-RAS International Conference on Humanoid Robots*, 2013, pp. 342–347.

[3] 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 *2016 IEEE-RAS International Conference on Humanoid Robots*, 2016, pp. 876–883.

[4] 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 *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2019, pp. 7294–7301.

[5] K. Kawaharazuka, K. Okada, and M. Inaba, "Characteristics, Management, and Utilization of Muscles in Musculoskeletal Humanoids: Empirical Study on Kengoro and Musashi," *Advanced Intelligent Systems*, 2026.

[6] S. C. Jacobsen, H. Ko, E. K. Iversen, and C. C. Davis, "Control strategies for tendon-driven manipulators," *IEEE Control Systems Magazine*, vol. 10, no. 2, pp. 23–28, 1990.

[7] K. Koganezawa and M. Yamazaki, "Mechanical stiffness control of tendon-driven joints," in *1999 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1999, pp. 818–825.

[8] H. Endo and M. Wada, "Reflex-like control for the coupled tendon-driven manipulator," in *1994 IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 3, 1994, pp. 1810–1815.

[9] H. G. Marques, F. Imtiaz, F. Iida, and R. Pfeifer, "Self-organization of reflexive behavior from spontaneous motor activity," *Biological Cybernetics*, vol. 107, no. 1, pp. 25–37, 2013.

[10] X. Liu, A. Rosendo, S. Ikemoto, M. Shimizu, and K. Hosoda, "Robotic investigation on effect of stretch reflex and crossed inhibitory response on bipedal hopping," *Journal of The Royal Society Interface*, vol. 15, no. 140, p. 20180024, 2018.

[11] K. Kawaharazuka, M. Nishiura, Y. Toshimitsu, Y. Omura, Y. Koga, Y. Asano, K. Okada, K. Kawasaki, and M. Inaba, "Robust Continuous Motion Strategy Against Muscle Rupture using Online Learning of Redundant Intersensory Networks for Musculoskeletal Humanoids," *Robotics and Autonomous Systems*, vol. 152, pp. 1–14, 2022.

[12] S. Gallagher, "Body Image and Body Schema: A Conceptual Clarification," *The Journal of Mind and Behavior*, vol. 7, no. 4, pp. 541–554, 1986.

[13] P. Haggard and D. M. Wolpert, "Disorders of Body Scheme," in *Higher-Order Motor Disorders, Ed. Freund, Jeannerod, Hallett, and Leiguarda*. Oxford University Press, 2005.

[14] Y. Nakanishi, K. Hongo, I. Mizuuchi, and M. Inaba, "Joint proprioception acquisition strategy based on joints-muscles topological maps for musculoskeletal humanoids," in *2010 IEEE International Conference on Robotics and Automation*, 2010, pp. 1727–1732.

[15] S. Ookubo, Y. Asano, T. Kozuki, T. Shirai, K. Okada, and M. Inaba, "Learning Nonlinear Muscle-Joint State Mapping Toward Geometric Model-Free Tendon Driven Musculoskeletal Robots," in *2015 IEEE-RAS International Conference on Humanoid Robots*, 2015, pp. 765–770.

[16] 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.

[17] 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.

[18] 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.

[19] Y. Yoshimitsu, T. Osa, and S. Ikemoto, "Forward/Inverse Kinematics Modeling for Tensegrity Manipulator Based on Goal-Conditioned Variational Autoencoder," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2023, pp. 6668–6673.

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

[21] P. W. Holland and R. E. Welsch, "Robust regression using iteratively reweighted least-squares," *Communications in Statistics – Theory and Methods*, vol. 6, no. 9, pp. 813–827, 1977.

[22] K. Kawaharazuka, M. Nishiura, Y. Koga, Y. Omura, Y. Toshimitsu, Y. Asano, K. Okada, K. Kawasaki, and M. Inaba, "Automatic Grouping of Redundant Sensors and Actuators Using Functional and Spatial Connections: Application to Muscle Grouping for Musculoskeletal Humanoids," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1981–1988, 2021.