IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

PIMBS: Efficient Body Schema Learning for Musculoskeletal Humanoids with Physics-Informed Neural Networks

Kento Kawaharazuka[1], Takahiro Hattori[1], Keita Yoneda[1], and Kei Okada[1]

***Abstract*— Musculoskeletal humanoids are robots that closely mimic the human musculoskeletal system, offering various advantages such as variable stiffness control, redundancy, and flexibility. However, their body structure is complex, and muscle paths often significantly deviate from geometric models. To address this, numerous studies have been conducted to learn body schema, particularly the relationships among joint angles, muscle tension, and muscle length. These studies typically rely solely on data collected from the actual robot, but this data collection process is labor-intensive, and learning becomes difficult when the amount of data is limited. Therefore, in this study, we propose a method that applies the concept of Physics-Informed Neural Networks (PINNs) to the learning of body schema in musculoskeletal humanoids, enabling high-accuracy learning even with a small amount of data. By utilizing not only data obtained from the actual robot but also the physical laws governing the relationship between torque and muscle tension under the assumption of correct joint structure, more efficient learning becomes possible. We apply the proposed method to both simulation and an actual musculoskeletal humanoid and discuss its effectiveness and characteristics.**

I. INTRODUCTION

Various musculoskeletal humanoids that closely mimic human anatomy have been developed [1]–[4]. These robots offer several advantages, such as enabling variable stiffness control through nonlinear elastic elements [5], [6] and allowing continued operation even when muscle rupture occurs due to muscle redundancy [7]. Furthermore, due to their human-like structure, they have been developed as systems capable of directly implementing and verifying human reflex control and learning mechanisms [8]–[10]. However, the complexity of their body structure often results in significant discrepancies between geometric models and the actual robot. The reasons for these discrepancies include variations in muscle paths and the inability of geometric models to accurately represent the muscles wrapped around the joints.

To address this issue, various methods have been developed to learn the body schema of musculoskeletal structures, which captures the relationship between joint angle, muscle tension, and muscle length from actual data. In [11], a method was proposed to construct a data table representing the relationship between joint angle and muscle length, which is updated using actual robot data. In [12], polynomial approximation was used instead of a data table, and by differentiating it to obtain the muscle Jacobian, joint angle estimation was performed. In [13], a neural network was used as an alternative to data tables and polynomial approximation, and it was applied to joint angle estimation and control. However, methods relying solely on joint angle and muscle length cannot account for factors such as wire elongation and nonlinear elastic elements. To address these limitations, other methods have been developed to learn the relationship between joint angle, muscle tension, and muscle length using actual robot data [14], [15]. On the other hand, obtaining a large amount of data from actual robots is challenging, making it desirable to develop methods that can accurately learn these relationships from a minimal amount of data.

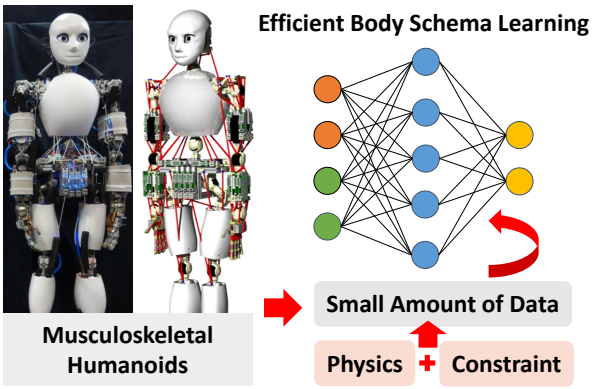

Fig. 1. The concept of this study: in musculoskeletal humanoids, efficient body schema learning can be achieved by leveraging a small amount of data obtained from the actual robot along with the physical knowledge and constraints of the musculoskeletal system.

In this study, we developed a method that enhances the neural network for the learning of the relationship between joint angle, muscle tension, and muscle length by incorporating constraints based on physical laws, enabling more efficient learning of the body schema (Fig. 1). We incorporate the concept of Physics-Informed Neural Networks (PINNs) [16], which embeds the relationship between muscle tension and joint torque – assuming that the joint geometric model is correct – into the loss function. We refer to this approach as the Physics-Informed Musculoskeletal Body Schema, or PIMBS. By utilizing this method, not only the data points themselves but also their differential information can be obtained, allowing for more efficient body schema learning from fewer data points. We conducted experiments using both a musculoskeletal structure in simulation and an actual musculoskeletal humanoid to evaluate the proposed method and discuss its characteristics.

[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, t-hattori, yoneda, k-okada]@jsk.imi.i.u-tokyo.ac.jp

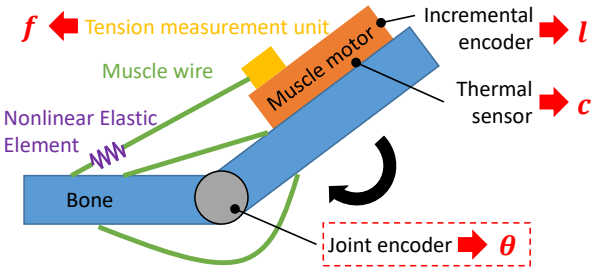

Fig. 2. The basic structure of musculoskeletal humanoids: muscle motors, equipped with incremental encoders, thermal sensors, and tension measurement units, are attached to the skeleton. Muscle wires extend from the muscle motors and are connected to the skeleton via nonlinear elastic elements. The presence of joint encoders depends on the specific robot.

The contributions of this study are summarized as follows.

- Proposal of PIMBS, a method for efficiently learning body schema from a small amount of data using physical laws
- Validation of the proposed method on a simulated musculoskeletal structure
- Application of the proposed method to an actual musculoskeletal humanoid and discussion of its characteristics

II. PIMBS: Efficient Body Schema Learning for Musculoskeletal Humanoids

A. Basic Structure of Musculoskeletal Humanoids

The general musculoskeletal structure is explained below. As shown in Fig. 2, the musculoskeletal structure typically consists of a skeleton, joints, and wires that simulate muscles. The wires are wound around pulleys attached to the motor, pass through the muscle tension measurement unit, and are connected to the skeleton. Nonlinear elastic elements are attached to the ends of the wires, enabling variable stiffness control. Muscle length l is obtained from the encoder attached to the motor, and muscle tension f is measured using a loadcell installed in the muscle tension measurement unit. Joint angle θ may or may not be directly measured depending on the actual robot. However, even when joint angle measurements are unavailable, they can be estimated with reasonable accuracy by combining muscle length variations with data from visual sensors [13].

When the wires do not stretch and nonlinear elasticity is absent, the following relationship holds using the function $\boldsymbol{h}$:

$$\boldsymbol{l} = \boldsymbol{h}(\boldsymbol{\theta}) \tag{1}$$

The derivative of this function with respect to $\boldsymbol{\theta}$ is called the muscle Jacobian and is denoted by $\boldsymbol{G}$:

$$\boldsymbol{G}(\boldsymbol{\theta}) = \frac{\partial \boldsymbol{h}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \tag{2}$$

Next, consider the case where the wires can stretch and nonlinear elastic elements are installed. Let the amount that the wire is stretched, which varies with joint angle and muscle tension, be denoted by $\Delta\boldsymbol{n}(\boldsymbol{\theta}, \boldsymbol{f})$. The following relationship holds:

$$\boldsymbol{l}^{geom} = \boldsymbol{h}^{geom}(\boldsymbol{\theta}) \tag{3}$$

$$\boldsymbol{l} = \boldsymbol{h}(\boldsymbol{\theta}, \boldsymbol{f}) = \boldsymbol{h}^{geom}(\boldsymbol{\theta}) + \Delta\boldsymbol{n}(\boldsymbol{\theta}, \boldsymbol{f}) \tag{4}$$

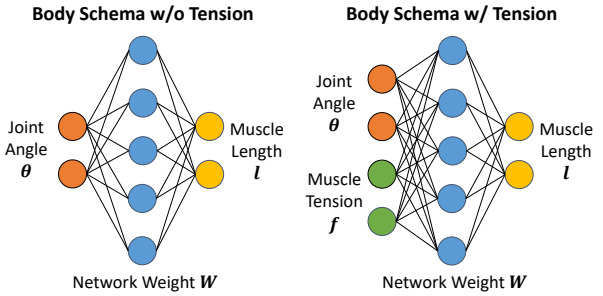

Fig. 3. The neural network structures for basic body schema learning: the input consists of joint angles, and in some cases, muscle tension; the output is the muscle length.

where $\boldsymbol{l}^{geom}$ represents the geometric wire length corresponding to the joint angle, and the actual muscle length $\boldsymbol{l}$ obtained from the motor is further adjusted by the amount $\Delta\boldsymbol{n}$. Note that the joint angle is included in $\Delta\boldsymbol{n}(\boldsymbol{\theta}, \boldsymbol{f})$ because, in general, the amount that the wire is stretched depends on the total wire length $\boldsymbol{l}^{geom}$. The muscle Jacobian is defined as the derivative of $\boldsymbol{l}^{geom}$ with respect to $\boldsymbol{\theta}$. However, since the variation of $\Delta\boldsymbol{n}$ with respect to $\boldsymbol{\theta}$ is generally much smaller than the variation of $\boldsymbol{h}^{geom}(\boldsymbol{\theta})$, the following approximation is often used:

$$\boldsymbol{G}(\boldsymbol{\theta}) = \frac{\partial \boldsymbol{h}^{geom}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \simeq \frac{\partial \boldsymbol{h}(\boldsymbol{\theta}, \boldsymbol{f})}{\partial \boldsymbol{\theta}} \tag{5}$$

B. Basic Body Schema Learning for Musculoskeletal Humanoids

The general method for learning the body schema of a musculoskeletal humanoid is described below. Basically, the body schema consists of a network that outputs muscle length from joint angle or from joint angle and muscle tension, as shown in Fig. 3. In other words, the network learns $\boldsymbol{h}$ in Eq. (1) or Eq. (4) as a neural network.

The learning process is simple. The actual robot is moved, and the data at each moment, $D_i = (\boldsymbol{\theta}^{data}i, \boldsymbol{f}^{data}i, \boldsymbol{l}_i^{data})$, is collected. The robot can be moved either by applying random muscle tension or by commanding muscle length based on a simple geometric model and adding muscle tension control to eliminate muscle slack and antagonistic effects. It is assumed that there is no muscle slack. It should be noted that the geometric model used here often differs significantly from the actual robot. The collected data for learning is denoted as D^{train}, with the total number of data points represented by N^{train}. Using these data, learning is performed with the mean squared error as the loss function, as shown below:

$$\boldsymbol{l}^{pred} = \boldsymbol{h}(\boldsymbol{\theta}^{data}) \quad \text{OR} \quad \boldsymbol{h}(\boldsymbol{\theta}^{data}, \boldsymbol{f}^{data}) \tag{6}$$

$$L_{basic} = \frac{1}{N^{train}} \sum (\boldsymbol{l}^{pred} - \boldsymbol{l}^{data})^2 \tag{7}$$

By using the trained $\boldsymbol{h}$, the robot can obtain the desired muscle length $\boldsymbol{l}$ by inputting either $\boldsymbol{\theta}$ or $(\boldsymbol{\theta}, \boldsymbol{f})$. Furthermore, muscle Jacobian $\boldsymbol{G}$, derived by differentiating $\boldsymbol{h}$, can be used for state estimation and torque control. The muscle length $\boldsymbol{l}$ is expressed as a relative change, with its value at $\boldsymbol{\theta} = \boldsymbol{0}$ set to $\boldsymbol{0}$.

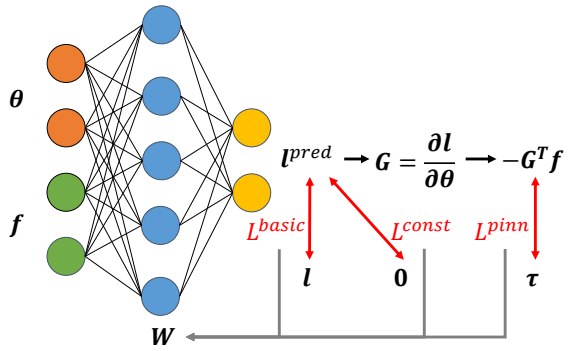

Fig. 4. The training scheme of Physics-Informed Musculoskeletal Body Schema, PIMBS.

This general body schema learning is highly effective, allowing for increasingly accurate learning as more data is collected. However, in practice, it is often difficult to obtain large amounts of data, making it desirable to learn with fewer data points. To address this, we incorporate the concept of Physics-Informed Neural Networks (PINNs), which leverage differential equations expressed through network derivatives. We refer to this network as the Physics-Informed Musculoskeletal Body Schema (PIMBS).

C. Physics-Informed Body Schema Learning for Musculoskeletal Humanoids

An overview of the training scheme of PIMBS is shown in Fig. 4. In body schema learning, it is generally assumed that the joint structure (i.e., the positional relationships between joints and links, as well as link weights) is correct, and only the relationship between joints and muscles is learned. Assuming the joint structure is correct, the gravity compensation torque $\boldsymbol{\tau}$ corresponding to the current joint angle $\boldsymbol{\theta}$ can be calculated. In musculoskeletal structures, the following relationship generally holds:

$$\boldsymbol{\tau} = -\boldsymbol{G}^T(\boldsymbol{\theta})\boldsymbol{f} \tag{8}$$

In other words, by utilizing $\boldsymbol{G}$, which is the derivative of the function $\boldsymbol{h}$ represented by the neural network, it is possible to introduce an additional loss function for learning. Therefore, the following loss can be defined:

$$\boldsymbol{G}^{pred} = \left.\frac{\partial \boldsymbol{h}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{data}} \quad \text{OR} \quad \left.\frac{\partial \boldsymbol{h}(\boldsymbol{\theta},\boldsymbol{f})}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{data},\boldsymbol{f}=\boldsymbol{f}^{data}} \tag{9}$$

$$L_{pinn} = \frac{1}{N^{train}}\sum(\boldsymbol{G}^{pred,T}\boldsymbol{f}^{data} + \boldsymbol{\tau}^{data})^2 \tag{10}$$

where $\boldsymbol{\tau}^{data}$ represents the gravity compensation torque for the joint angle $\boldsymbol{\theta}^{data}$. Additionally, constraint conditions can also be described. When $\boldsymbol{\theta} = \boldsymbol{0}$ and $\boldsymbol{f} = \boldsymbol{0}$, $\boldsymbol{l}$ should be $\boldsymbol{0}$, allowing for the definition of the following loss:

$$\boldsymbol{l}^{pred} = \boldsymbol{h}(\boldsymbol{0}) \quad \text{OR} \quad \boldsymbol{h}(\boldsymbol{0},\boldsymbol{0}) \tag{11}$$

$$L_{const} = (\boldsymbol{l}^{pred} - \boldsymbol{0})^2 \tag{12}$$

By additionally utilizing these losses L_{pinn} and L_{const} alongside L_{basic}, the body schema can be learned more efficiently. For the experiments, we compare the learning performance by varying the loss function L_{train} among L_{basic} (**Basic**), $L_{basic} + L_{const}$ (**Basic+Const**), $L_{basic} + \alpha L_{pinn}$ (**Basic+PINN**), and $L_{basic} + L_{const} + \alpha L_{pinn}$ (**Basic+Const+PINN**). Here, α is the weight for the physical laws. Among these loss functions, **Basic** is the conventional approach that has been used in almost all previous studies. Although there are other methods such as polynomial approximation and data table-based approaches, they are not included as baselines in this study due to the difficulty of making a fair comparison with neural networks. Note that all displayed loss values are scaled by a factor of 1.0×10^5 for clearer visualization.

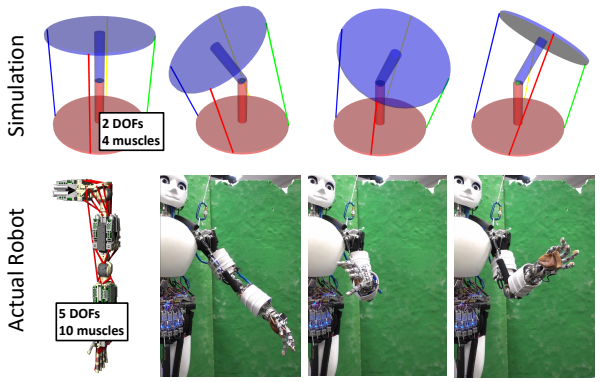

Fig. 5. The musculoskeletal structures handled in our experiments: 2-DOF 4-muscle musculoskeletal simulation and 5-DOF 10-muscle left arm of the musculoskeletal humanoid.

III. Experiments

A. Experimental Setup

The experimental setup of this study is described here. In this study, we use a 2-DOF 4-muscle musculoskeletal simulation, and the 5-DOF 10-muscle left arm of a musculoskeletal humanoid, as shown in Fig. 5.

First, for the 2-DOF 4-muscle simulation, experiments are conducted using two formulations defined by Eq. (1) and Eq. (4). To simplify understanding, the formulation in Eq. (1) is referred to as the AL-Map (angle-length map), and the one in Eq. (4) is referred to as the ATL-Map (angle-tension-length map). Initially, training data for $(\boldsymbol{\theta}, \boldsymbol{f}, \boldsymbol{l}, \boldsymbol{\tau})$ is generated in simulation. The joint angle is randomly varied within the range of $[-0.5, 0.5]$ [rad]. Once $\boldsymbol{\theta}$ is determined, the geometric muscle length $\boldsymbol{l}^{geom}$ and the gravity compensation torque $\boldsymbol{\tau}$ are calculated geometrically. During this process, the stretch Δn of each nonlinear elastic element is obtained using the following relationship, and $\boldsymbol{l}$ is then calculated in the same form as Eq. (4):

$$f = e^{K\Delta n} - 1 \tag{13}$$

where K is a coefficient, set to 1000 in this study. The muscle tension $\boldsymbol{f}$ is generated by solving the following quadratic programming problem:

$$\begin{aligned} \underset{\boldsymbol{f}}{\text{minimize}} \quad & \boldsymbol{f}^T\boldsymbol{W}\boldsymbol{f} \\ \text{subject to} \quad & \boldsymbol{\tau} = -\boldsymbol{G}^T(\boldsymbol{\theta})\boldsymbol{f} \\ & \boldsymbol{f} \geq \boldsymbol{f}^{min} \end{aligned} \tag{14}$$

where W represents the weighting matrix, which is set to the identity matrix in this study. Additionally, f^{min} represents the minimum muscle tension. In experiments related to the AL-Map, f^{min} is set to a fixed value of 0 [N]. This allows $\boldsymbol{\tau}$, $\boldsymbol{f}$, and $\boldsymbol{l}$ to be uniquely determined from $\boldsymbol{\theta}$, enabling the construction of a body schema in the form of the AL-Map. The formulation in Eq. (1) assumes inextensible wires, but a similar formulation can be applied even when wires are stretched, as long as $\boldsymbol{l}$ can be uniquely determined from $\boldsymbol{\theta}$. On the other hand, in experiments related to the ATL-Map, variations in $\boldsymbol{f}$ are also accepted, allowing f^{min} to be randomly varied within the range of [10, 300] N during the calculation of $\boldsymbol{f}$. In this way, experiments are conducted on a more complex body schema that takes both $\boldsymbol{\theta}$ and $\boldsymbol{f}$ as inputs.

The following experiments were conducted using the left arm of a musculoskeletal humanoid [4], which has 5 joints and 10 muscles. The joint angle of the left arm was controlled and held stationary repeatedly in a random manner based on a geometric model. During stationary phases, muscle tension control was applied to eliminate muscle slack, and data was collected in this process. Since each muscle is equipped with a nonlinear elastic element, the body schema was learned in the form of the ATL-Map.

Details of the body schema learning are as follows. The body schema is constructed using a neural network with three fully connected layers. The input dimension is N or $N+M$, the number of hidden units is 1000, and the output dimension is M, where N represents the number of joints and M represents the number of muscles. A hyperbolic tangent function is applied after the first and second fully connected layers. The batch size is set to match N^{train}, the number of epochs is 20000, and Adam [17] is used as the update rule. The weight α for the physical laws is fixed at 1.0×10^{-5} in simulation, while in the actual robot experiments, it is varied among $1.0 \times 10^{\{-5,-6,-7,-8\}}$ (details will be discussed in Section III-D). Experiments were performed by varying the number of training data N^{train} and the loss function, between **Basic**, **Basic+Const**, **Basic+PINN**, and **Basic+Const+PINN**. In simulations, a separate evaluation dataset D^{eval} consisting of 1000 randomly generated points was prepared. The evaluation value L_{eval} is defined as the result of calculating Eq. (7) using D^{eval}. In the actual robot experiments, out of 496 data points obtained from the actual robot, the data excluding N^{train} points was used as D^{eval}, and L_{eval} was calculated. The final model selected is the one with the minimum L_{basic} obtained from the training data during all epochs, and the corresponding evaluation value is denoted as L_{eval}^{best}. In this study, the performance of different learning methods is compared based on the change in L_{eval} during the learning process and the final value of L_{eval}^{best}.

B. Body Schema Learning of AL-Map for 2-DOF 4-muscle Musculoskeletal Simulation

First, we describe the results of learning the body schema represented by the AL-Map in a 2-DOF 4-muscle musculoskeletal simulation. Here, experiments are conducted by

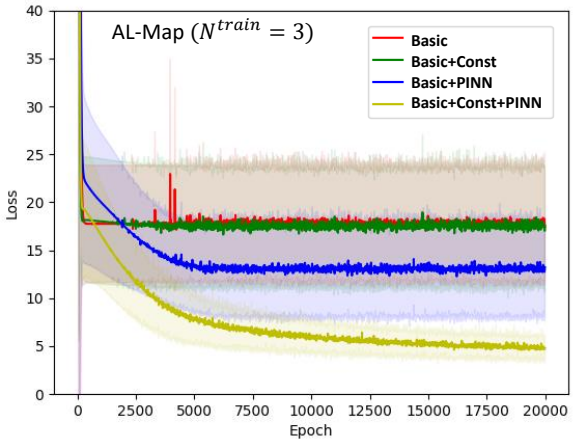

Fig. 6. The transitions of L_{eval} when training AL-Map with 3 data points in the 2-DOF 4-muscle musculoskeletal simulation.

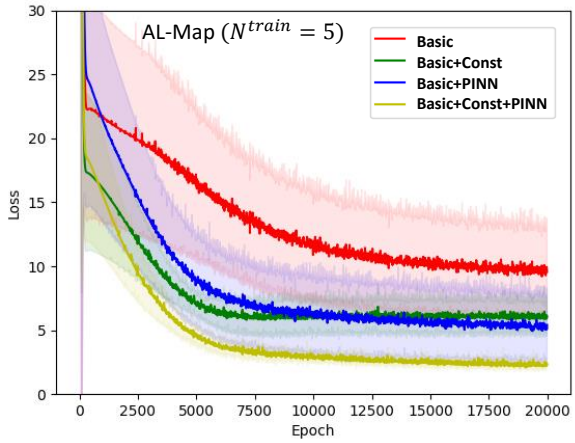

Fig. 7. The transitions of L_{eval} when training AL-Map with 5 data points in the 2-DOF 4-muscle musculoskeletal simulation.

varying $N^{train} = \{3, 5, 10\}$. The results for $N^{train} = 3$, $N^{train} = 5$, and $N^{train} = 10$ are shown in Fig. 6, Fig. 7, and Fig. 8, respectively. Each experiment was repeated five times with different random seeds, and the average and standard deviation of the transition of L_{eval} are presented. This applies to all subsequent graphs as well.

For $N^{train} = 3$, **Basic** and **Basic+Const** show similar performance, while both **Basic+PINN** and **Basic+Const+PINN** exhibit significant performance improvement. For $N^{train} = 5$, **Basic+Const** outperforms **Basic**, and both **Basic+PINN** and **Basic+Const+PINN** further enhance performance. For $N^{train} = 10$, these differences become negligible. As a general trend, increasing N^{train} improves overall performance, but the performance gap between methods decreases. Notably, for $N^{train} = 5$, both **Const** and **PINN** contribute to performance improvement, with **Basic+Const+PINN** achieving the highest performance.

Table I shows the average and standard deviation of L_{eval}^{best} for each learning method. For $N^{train} = \{3, 5\}$, **Basic+Const+PINN** achieves the best performance, reducing L_{eval}^{best} by approximately 60–70% compared to **Basic**. In contrast, for $N^{train} = 10$, performance differences become minimal. Additionally, it is evident that **PINN** has a greater

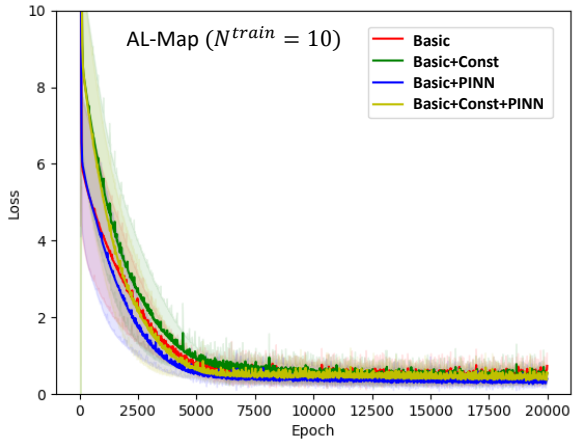

Fig. 8. The transitions of L_{eval} when training AL-Map with 10 data points in the 2-DOF 4-muscle musculoskeletal simulation.

impact on performance than **Const**.

Fig. 9 shows the relationship between joint angle and muscle length for $N^{train} = 5$. The relationship in the geometric model to be learned is labeled as Geometric, while the relationship derived from the learned body schema is labeled as Trained, for comparison. The left figure shows the transition when the first joint is moved, and the right figure shows the transition when the second joint is moved. When **Const** is included in the loss function, the muscle length is learned to become zero near the origin of the joint angle, with the transitions of each muscle intersecting neatly at the origin. On the other hand, more errors are observed without **Const** compared to when it is included. When **PINN** is included in the loss function, performance is improved, particularly at the extremes of the joint angle range. Since the joint angle range is randomly varied, data near the extremes are sparse, but utilizing gradient information enables successful extrapolation in these regions.

TABLE I

COMPARISON OF L_{eval}^{best} AND ITS VARIANCE FOR FOUR TRAINING METHODS OF AL-MAP IN THE 2-DOF 4-MUSCLE MUSCULOSKELETAL SIMULATION.

Method	$N^{train} = 3$	$N^{train} = 5$	$N^{train} = 10$
Basic	17.75 ± 6.10	9.70 ± 3.60	0.44 ± 0.21
Basic+Const	17.44 ± 6.36	5.98 ± 1.37	0.42 ± 0.19
Basic+PINN	12.96 ± 5.04	5.72 ± 3.41	**0.29 ± 0.12**
Basic+Const+PINN	**6.99 ± 1.86**	**2.73 ± 0.38**	0.47 ± 0.23

C. Body Schema Learning of ATL-Map for 2-DOF 4-muscle Musculoskeletal Simulation

Next, the results of learning the body schema represented by the ATL-Map in a 2-DOF 4-muscle musculoskeletal simulation are presented. Here, experiments are conducted by varying $N^{train} = \{5, 10, 30\}$. Since the learning is more challenging than that of AL-Map, the amount of data is set to be greater than in the case of AL-Map. The results for $N^{train} = 5$, $N^{train} = 10$, and $N^{train} = 30$ are shown in Fig. 10, Fig. 11, and Fig. 12, respectively.

For $N^{train} = 5$, the performance is ranked in descending order as follows: **Basic+Const+PINN**, **Basic+Const**, **Basic+PINN**, and **Basic**. For $N^{train} = 10$, a similar trend is observed. For $N^{train} = 30$, **Basic+Const+PINN**, **Basic+Const**, and **Basic+PINN** show almost the same performance, with only **Basic** showing a slight drop in performance. Overall, both **Const** and **PINN** have a certain positive effect, and the combined method **Basic+Const+PINN** achieves the highest performance.

Table II shows the mean and standard deviation of L_{eval}^{best} for each learning method. In all cases of $N^{train} = \{5, 10, 30\}$, **Basic+Const+PINN** achieves the highest performance, with L_{eval}^{best} reduced by approximately 20–35% compared to **Basic**. The performance gap decreases as N^{train} increases. Unlike in the case of the AL-Map, it is also evident that **Const** contributes more to performance improvement than **PINN**.

TABLE II

COMPARISON OF L_{eval}^{best} AND ITS VARIANCE FOR FOUR TRAINING METHODS OF ATL-MAP IN THE 2-DOF 4-MUSCLE MUSCULOSKELETAL SIMULATION.

Method	$N^{train} = 5$	$N^{train} = 10$	$N^{train} = 30$
Basic	783.45 ± 114.46	176.25 ± 34.74	14.92 ± 3.85
Basic+Const	623.38 ± 37.68	146.55 ± 35.31	12.37 ± 3.45
Basic+PINN	706.89 ± 197.87	152.92 ± 32.71	12.49 ± 2.76
Basic+Const+PINN	**515.84 ± 95.48**	**132.35 ± 20.99**	**11.91 ± 2.89**

D. Body Schema Learning of ATL-Map for the Actual Musculoskeletal Humanoid

Finally, we describe the results of learning the body schema represented by the ATL-Map on the 5-DOF 10-muscle left arm of a musculoskeletal humanoid [4]. Experiments were conducted by varying $N^{train} = \{10, 30\}$. In addition to the results obtained with $\alpha = 1.0 \times 10^{-5}$ – as used in simulation experiments – we also show results for **Basic+Const+PINN** with $\alpha = 1.0 \times 10^{\{-6,-7,-8\}}$. For the case of $\alpha = 1.0 \times 10^{-5}$, the results for $N^{train} = 10$ and 30 are shown in the upper parts of Fig. 13 and Fig. 14, respectively. For **Basic+Const+PINN** with $\alpha = 1.0 \times 10^{\{-5,-6,-7,-8\}}$, the results for $N^{train} = 10$ and 30 are shown in the lower parts of Fig. 13 and Fig. 14, respectively.

First, for $\alpha = 1.0 \times 10^{-5}$, regardless of whether $N^{train} = 10$ or 30, the performance is highest for **Basic+Const**, followed by **Basic**, while adding **PINN** leads to a degradation in performance over training epochs. In contrast, when using smaller values of $\alpha = 10^{\{-6,-7,-8\}}$, this performance degradation is gradually mitigated.

Table III shows the average and standard deviation of L_{eval}^{best} for each learning method. When $N^{train} = 10$, **Basic+Const+PINN** with $\alpha = 10^{-7}$ achieves the best performance. When $N^{train} = 30$, **Basic+Const+PINN** with $\alpha = 10^{-8}$ performs best. Furthermore, a consistent trend is observed in which the performance gap decreases as N^{train} increases.

IV. DISCUSSION

The experimental results obtained in this study are summarized here. Regarding the simulation experiments, both **Const** and **PINN** demonstrated a certain level of effectiveness, and

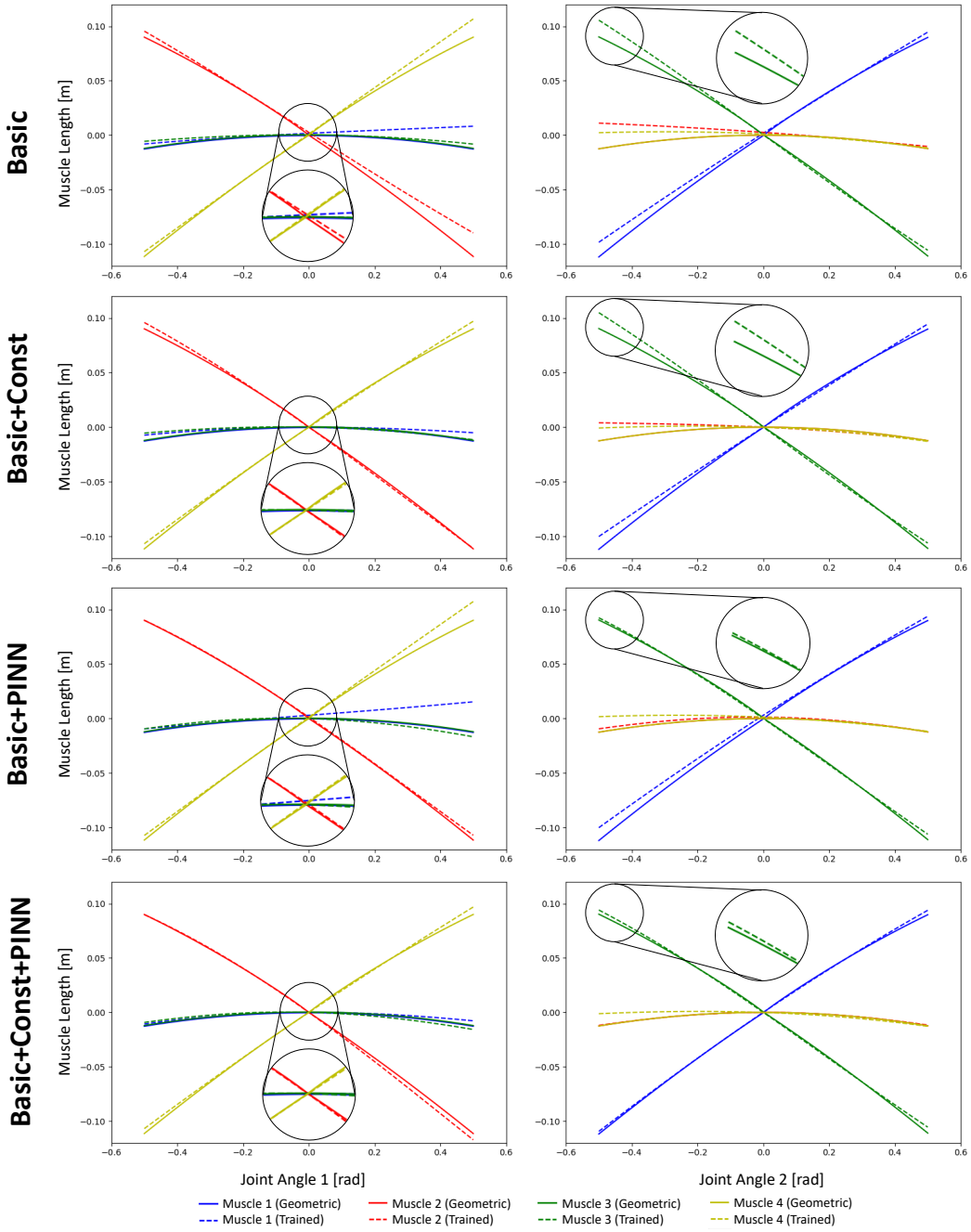

Fig. 9. The relationship between joint angle and muscle length when training AL-Map with 5 data points in the 2-DOF 4-muscle musculoskeletal simulation.

it was found that the combination of these two methods, **Basic+Const+PINN**, achieved the highest performance. Which of **PINN** or **Const** has a greater effect depends on the problem setting. This difference is more pronounced when N^{train} is small, and as N^{train} increases, the performance gap becomes smaller. In contrast, for the actual robot experiments, applying the same weight α for the physical laws as used in simulation led to a degradation in performance when **PINN** was added. However, by reducing α, this performance degradation was mitigated, and **Basic+Const+PINN** once again achieved the highest performance, similar to the simulation results. Notably, in simulation, changing the value of α did not result in significant differences in performance.

Several insights were gained from the experiments. By introducing the concept of PINN and utilizing the gradient information within the network, it was possible to ensure a certain level of performance at the boundaries of joint angle limits where the amount of data is scarce. When combined with the constraint on returning to the origin imposed by **Const**, both quantitative and qualitative performance improvements were observed. Also, the extent to which physical laws are considered has little effect in simulations, where there is no discrepancy between the data and physical laws. However, in the actual robot experiments, factors like

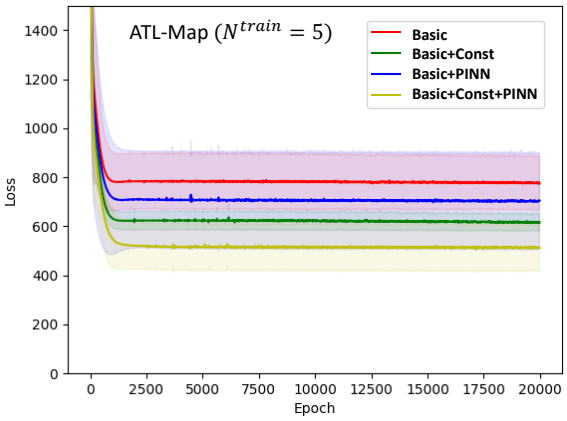

Fig. 10. The transitions of L_{eval} when training ATL-Map with 5 data points in the 2-DOF 4-muscle musculoskeletal simulation.

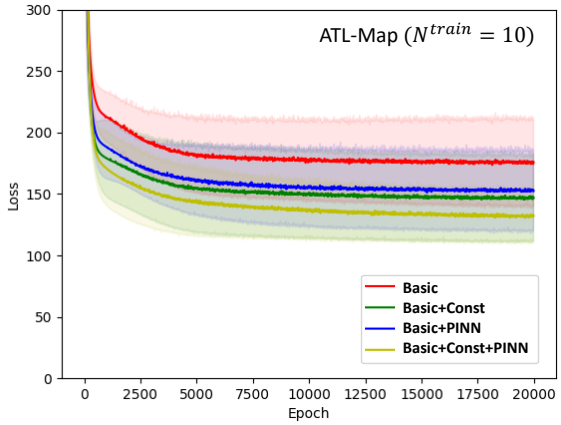

Fig. 11. The transitions of L_{eval} when training ATL-Map with 10 data points in the 2-DOF 4-muscle musculoskeletal simulation.

friction – difficult to model as static physical laws – introduce inconsistencies between the collected data and physical laws. Therefore, when physical laws are weighted as heavily as in simulation, the model is overly influenced by incorrect assumptions, leading to a significant drop in performance. Reducing the weight for the physical laws helps suppress this degradation while still benefiting somewhat from gradient-based regularization. However, even with this adjustment, performance in the actual robot experiments did not reach the level observed in simulation.

The future outlook for this research is discussed below. Although this study demonstrated a certain level of effectiveness, when the influence of friction is significant, such

TABLE III

COMPARISON OF L_{eval}^{best} AND ITS VARIANCE FOR FOUR TRAINING METHODS OF ATL-MAP IN THE LEFT ARM OF THE MUSCULOSKELETAL HUMANOID.

Method	$N^{train} = 10$	$N^{train} = 30$
Basic	299.42 ± 70.53	19.41 ± 4.84
Basic+Const	256.57 ± 72.48	18.99 ± 4.38
Basic+PINN ($\alpha = 10^{-5}$)	423.32 ± 165.67	43.16 ± 15.87
Basic+Const+PINN ($\alpha = 10^{-5}$)	449.73 ± 217.19	44.27 ± 11.03
Basic+Const+PINN ($\alpha = 10^{-6}$)	409.95 ± 212.65	22.65 ± 4.88
Basic+Const+PINN ($\alpha = 10^{-7}$)	276.16 ± 119.48	**18.17 ± 4.79**
Basic+Const+PINN ($\alpha = 10^{-8}$)	**235.31 ± 65.11**	18.39 ± 4.51

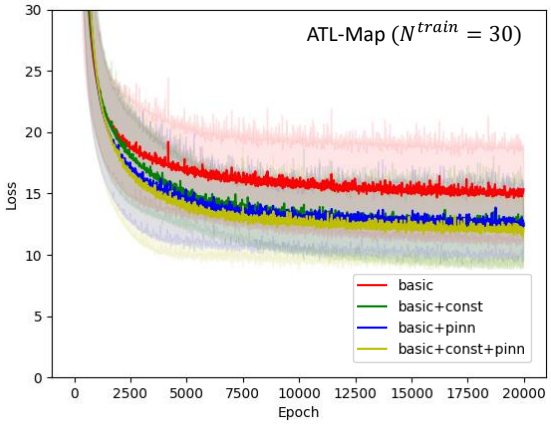

Fig. 12. The transitions of L_{eval} when training ATL-Map with 30 data points in the 2-DOF 4-muscle musculoskeletal simulation.

as in the actual robot, modeling that accounts for this effect is crucial. In the current static modeling framework, it is difficult to account for frictional effects that depend on the direction of motion. Although we attempted a simplified model that considers transmission efficiency, this did not lead to substantial improvements in performance. Ideally, a dynamic model that incorporates friction, its resulting dynamics, and hysteresis effects should be constructed. However, this would require a greater amount of data, necessitating careful consideration of the balance between modeling complexity and data availability. Furthermore, in this study, it was assumed that the joint model was accurate, but in many cases, this assumption may not hold. Therefore, future work will aim to develop a formulation capable of handling errors in the joint model as well.

V. CONCLUSION

In this study, we proposed a new learning method called Physics-Informed Musculoskeletal Body Schema (PIMBS), which incorporates the concept of Physics-Informed Neural Networks (PINNs) for more efficient body schema learning in musculoskeletal humanoids. This method leverages not only joint angle, muscle tension, and muscle length data but also the muscle Jacobian obtained through differentiation, as well as the relationship between muscle tension and gravity compensation torque, assuming that the joint structure is correct. We demonstrate that this approach enables more efficient body schema learning from a small amount of data with higher performance than conventional methods in both simulation and the actual musculoskeletal humanoid. The performance improvement is more pronounced when the amount of data is limited. On actual robots, friction can reduce performance; however, we found that adjusting the weight for the physical laws helped mitigate this issue to some extent. In this study, we focused on learning the static relationship between joint angle, muscle tension, and muscle length. In the future, we aim to extend this to dynamic relationships, incorporating the effects of wire friction, viscosity, and dynamics.

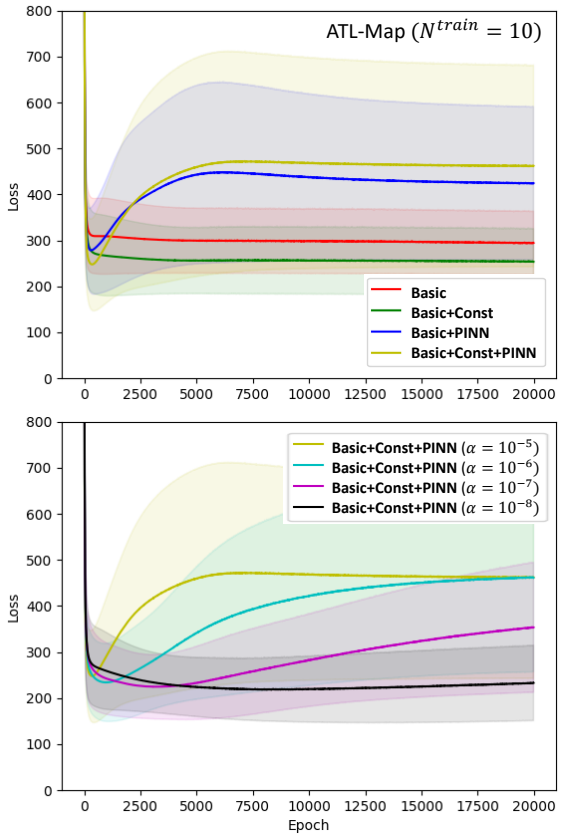

Fig. 13. The transitions of L_{eval} when training ATL-Map with 10 data points in the left arm of the musculoskeletal humanoid.

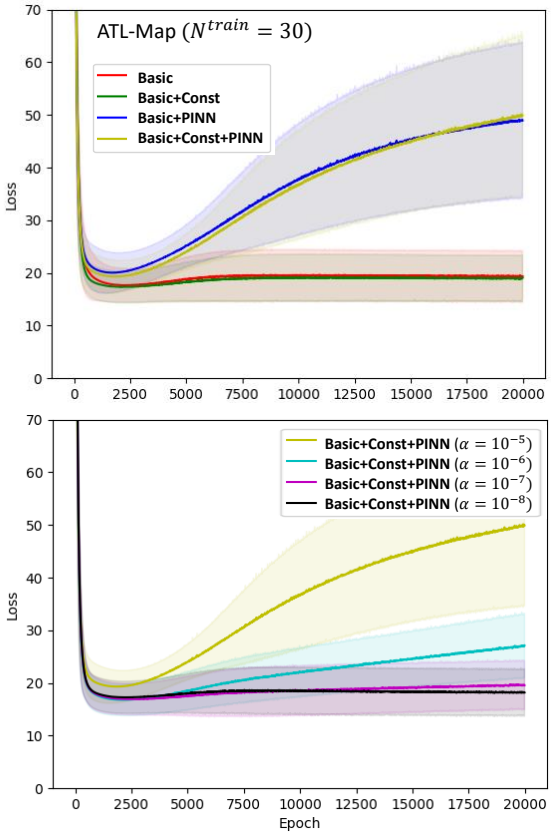

Fig. 14. The transitions of L_{eval} when training ATL-Map with 30 data points in the left arm of the musculoskeletal humanoid.

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

[5] 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.

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

[7] 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.

[8] H. Endo and M. Wada, "Reflex-like control for the coupled tendon-driven manipulator," in *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] Y. Nakanishi, K. Hongo, I. Mizuuchi, and M. Inaba, "Joint proprioception acquisition strategy based on joints-muscles topological maps for musculoskeletal humanoids," in *IEEE International Conference on Robotics and Automation*, 2010, pp. 1727–1732.

[12] 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 *IEEE-RAS International Conference on Humanoid Robots*, 2015, pp. 765–770.

[13] 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.

[14] 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.

[15] 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.

[16] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational physics*, vol. 378, pp. 686–707, 2019.

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