IEEE Robotics and Automation Letters (RAL) paper presented at the
2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

# Estimation and Control of Motor Core Temperature with Online Learning of Thermal Model Parameters: Application to Musculoskeletal Humanoids

Kento Kawaharazuka[1], Naoki Hiraoka[1], Kei Tsuzuki[1], Moritaka Onitsuka[1], Yuki Asano[1], Kei Okada[1], Koji Kawasaki[2], and Masayuki Inaba[1]

***Abstract*— The estimation and management of motor temperature are important for the continuous movements of robots. In this study, we propose an online learning method of thermal model parameters of motors for an accurate estimation of motor core temperature. Also, we propose a management method of motor core temperature using the updated model and anomaly detection method of motors. Finally, we apply this method to the muscles of the musculoskeletal humanoid and verify the ability of continuous movements.**

## I. INTRODUCTION

Managing motor temperature is important for the continuous movements of robots. Especially, the equipment of motors with sufficient specifications is difficult for the life-sized humanoid robot in order to keep its human-like proportion and weight. Thus, joint torque must be limited to keep the motor temperature within the rated value.

Several management methods of motor temperature have been developed from a variety of perspectives so far. As a method to manage motor temperature by hardware, a water cooling method of motors has been proposed [1] and the developed robot can jump high with a momentary large current. Also, motor cooling methods using a heat sink, the air, and phase change materials have been proposed [2]. However, these additional hardware is limited for the humanoid robots in order to keep their human-like specifications.

Common methods to constrain joint torque are minimizing joint torque by optimization [3] and using inequality constraints to set the maximum value of joint torque when optimizing [4]. However, because these methods do not directly handle motor temperature, they cannot guarantee that motor temperature is lower than the rated value. Although there is a method estimating motor housing temperature and moving the center of gravity for thermal relaxation [5] as an example to directly consider the motor temperature in the cost function at optimization, the cost function also cannot guarantee the rated value.

Compared with these methods, Urata, et al. have estimated motor core temperature from motor housing temperature and electric current, and developed a method to calculate the maximum current that can flow during an extremely short amount of time, and constrain a current by the maximum value [6]. Also, Kumagai, et al. have proposed a method to estimate motor housing temperature and constrain joint torque by the maximum value which can be applied for 120 seconds [7].

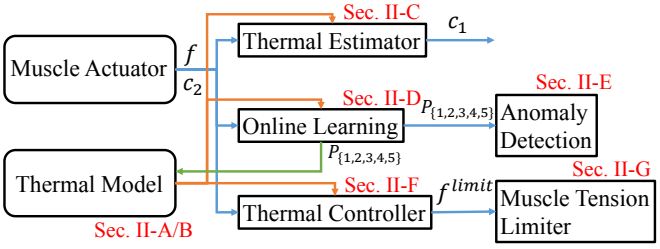


Fig. 1: System overview.

However, there are several problems in the proposed software methods to control the motor temperature. First, although there are some parameters for the thermal estimation of motors, among these parameters, the assumption that the ambient temperature is constant is often wrong. Regarding motors in the body parts that accumulate heat easily, the ambient temperature changes dynamically. Also, thermal and terminal resistances of motors gradually change due to the poor use of the robot or deterioration over time. Although identification of thermal model parameters is conducted in [7], the identification is conducted offline and only motor housing temperature is considered by simplifying the thermal model. Second, there is a problem in the control of motor temperature. The proposed methods so far have restricted motor output by the maximum current or joint torque that guarantees the rated value when applying the constant current or joint torque during a certain period. Because [6] considers an extremely short amount of time, although it can be applied to instantaneous motions like a jump, it is not suitable for ordinary motions. Because [7] does not change the maximum output dynamically, a value lower than the actual possible value is used as the maximum value.

Therefore, we propose an online learning method of thermal model parameters for the estimation of accurate motor core temperature, and the accurate control of the temperature using the updated model and dynamic optimization. Also, we conduct anomaly detection of motors using the change in thermal model parameters. In this study, we verify the effectiveness by applying this method to the musculoskeletal humanoid. Because the musculoskeletal humanoid [8], [9]

[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, hiraoka, tsuzuki, onitsuka, asano, k-okada, inaba]@jsk.t.u-tokyo.ac.jp
[2] The author is associated with TOYOTA MOTOR CORPORATION. koji_kawasaki@mail.toyota.co.jp

TABLE I: Notations in this paper

| Notation | Definition |
|---|---|
| $c_1$ | motor core temperature [°C] |
| $c_1^{max}$ | maximum motor core temperature to be limited [°C] |
| $c_2$ | motor housing temperature [°C] |
| $c_a$ | ambient temperature [°C] |
| $f$ | muscle tension [N] |
| $f^{limit}$ | maximum muscle tension to be limited [N] |
| $l^{ref}$ | target muscle length [mm] |
| $\Delta l$ | muscle elongation value [mm] |
| $\bullet_k$ | the value at the time step $k$ |
| $\bullet_{[k^{from},k^{to}]}$ | the value sequence at the time steps from $k^{from}$ to $k^{to}$ |
| $P_{\{1,2,3,4,5\}}$ | thermal parameters to be updated |
| $P_{\{1,2,3,4,5\},sim}$ | thermal parameters of simulated muscle actuator |

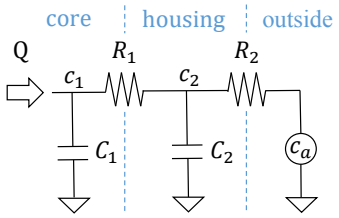


Fig. 2: Overview of a basic two-resistor thermal model.

mimics the human body in detail and has many muscles, the restrictions of weight and proportion are strict, and air or water cooling of motors is difficult to apply (on the other hand, there is a study on sweating robots [10]). Since the improvement of hardware is difficult, thermal restriction by software is suitable. The actuator of the musculoskeletal humanoid used in this study is not a pneumatic actuator but a motor which winds a muscle wire by a pulley. When applying this study to ordinary axis-driven humanoids, muscle tension should be converted to joint torque.

The contributions of this study are shown below.

- Online learning of thermal model parameters for an accurate estimation of motor core temperature
- Dynamic calculation of maximum output of motors using the updated thermal model and optimization
- Anomaly detection of motors using the change in thermal model parameters
- Verification of continuous motions of the musculoskeletal humanoid by applying this method to a muscle length-based control

## II. Proposed Method

First, we will explain a basic two-resistor thermal model to estimate motor core temperature. After that, we will construct a model with thermal parameters as variables, and explain thermal estimation of motors, online learning of the thermal parameters, anomaly detection of motors, calculation of maximum output, and control method to restrict muscle tension. The entire system and corresponding sections are shown in Fig. 1, and notations are shown in Table. I.

### A. Basic Thermal Model

In this study, we use a two-resistor thermal model as shown in Fig. 2, which is the same model as stated in [6]. This model is applicable to various classical motors such as

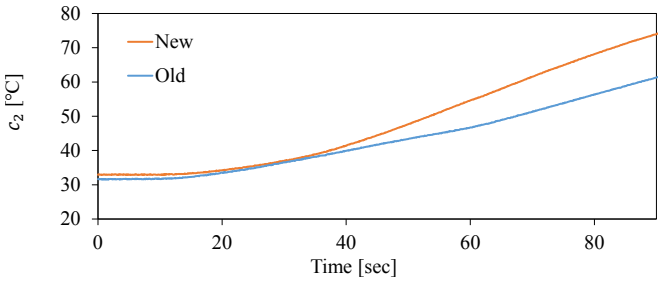


Fig. 3: Difference of transition of $c_2$ when applying $f = 200$ [N] between new and old motors

brushless and brushed DC motors. We assume heat capacity $C_1, C_2$ for motor core and housing, thermal resistance $R_1$ between motor core and housing, and thermal resistance $R_2$ between motor housing and ambient temperature. There is a relationship among $c_1$, $c_2$, and $c_a$, as shown below,

$$C_1 \frac{dc_1}{dt} = Q - \frac{c_1 - c_2}{R_1} \tag{1}$$

$$C_2 \frac{dc_2}{dt} = \frac{c_1 - c_2}{R_1} - \frac{c_2 - c_a}{R_2} \tag{2}$$

$$Q = R_e i^2 \tag{3}$$

where $R_e$ is a wire-wound resistance and $i$ is an electric current that flows through the motor core. Also, in order to apply this model to the musculoskeletal humanoid, we convert $i$ to $f$ by the equation below,

$$D_{pulley} f = E_{gear} D_{gear} E_{motor} K_t i \tag{4}$$

where $K_t$ is a torque constant, $E_{\{motor,gear\}}$ is a transmission efficiency of the motor or gear, $D_{gear}$ is a gear ratio, and $D_{pulley}$ is a radius of the pulley. We organize Eq. 1 and Eq. 2 with $K = R_e (D_{pulley}/(E_{gear} D_{gear} E_{motor} K_t))^2$, as shown below,

$$\dot{c}_1 = \frac{K}{C_1} f^2 - \frac{c_1 - c_2}{R_1 C_1} \tag{5}$$

$$\dot{c}_2 = \frac{c_1 - c_2}{R_1 C_2} - \frac{c_2 - c_a}{R_2 C_2} \tag{6}$$

where $\dot{c}_{\{1,2\}}$ represents $dc_{\{1,2\}}/dt$. $c_1$ can be calculated by discretely repeating the recurrence relations of Eq. 5 and Eq. 6. Also, in the usual case that $c_2$ can be obtained from a thermal sensor attached to motor housing, $c_1$ can be calculated just by repeating Eq. 5.

However, the actual thermal parameters are different from the parameters obtained from datasheets. This is considered to be because of heat dissipation to the attached metal parts, error of the ambient temperature, deterioration or burnout of motors, etc. As one example, we compared a new motor and an old motor used for over half a year, whose rotation is inferior. We show the difference of the transition of $c_2$ when applying $f = 200$ [N] over 90 seconds in Fig. 3. $c_2$ of the two motors were different by about 15 °C in 90 seconds. Because $c_1$ is more sensitive than $c_2$, a larger difference must be generated. Thus, the thermal model should always be updated, and we propose the method below.

### B. Proposed Thermal Model

We construct a model shown below, by setting the parameters in Eq. 5 and Eq. 6 as variables,

$$\dot{c}_1 = W_1 \exp(P_1) f^2 - \frac{c_1 - c_2}{W_2 \exp(P_2)} \tag{7}$$

$$\dot{c}_2 = \frac{c_1 - c_2}{W_3 \exp(P_3)} - \frac{c_2 - W_5(1+P_5)}{W_4 \exp(P_4)} \tag{8}$$

where $W_1 = K/C_1$, $W_2 = R_1 C_1$, $W_3 = R_1 C_2$, $W_4 = R_2 C_2$, and $W_5 = c_a$. $P_{\{1,2,3,4,5\}}$ are parameters updated in this study, and $\exp(\cdot)$ is an exponential function. The reason why the exponential function is applied to $P_{\{1,2,3,4\}}$ is because $W_{\{1,2,3,4\}} \exp(P_{\{1,2,3,4\}})$ must be positive. When $P_{\{1,2,3,4,5\}} = 0$, this model is equivalent with Eq. 5 and Eq. 6. We can interpret $P_1$ as a coefficient to determine the quantity of heat from the current, $P_2$ as a time constant of heat escaping from the motor core to the motor housing, $P_3$ as a time constant of heat flowing in the motor housing from the motor core, $P_4$ as a time constant of heat escaping from the motor housing to the ambient, and $P_5$ as a coefficient expressing the ambient temperature ($W_5(1+P_5)$ expresses the updated ambient temperature).

We can simply express this model by using the function of $h_1$ and $h_2$, as shown below.

$$\dot{c}_1 = h_1(f, c_1, c_2) \tag{9}$$

$$\dot{c}_2 = h_2(c_1, c_2) \tag{10}$$

In this study, we update $P_{\{1,2,3,4,5\}}$ to match the actual parameters of the robot online for the accurate thermal estimation and control using Eq. 9 and Eq. 10.

### C. Thermal Estimator: Estimation of Motor Core Temperature

In this study, as a common configuration of motors, we assume that a thermal sensor is attached to the motor housing. In this case, a method to estimate $c_1$ is simple. We determine an update interval $\Delta t_{est}$ and estimate $c_1$ as below,

$$c_{1,t+1} = c_{1,t} + h_1(f_t, c_{1,t}, c_{2,t}) \Delta t_{est} \tag{11}$$

where $\{c_1, c_2, f\}_t$ is $\{c_1, c_2, f\}$ at the current time step $t$.

In this study, we set $\Delta t_{est} = 0.02$ [sec].

### D. Online Learning of Thermal Model

By using the obtained sensor data of $c_2$ and $f$, we update $P_{\{1,2,3,4,5\}}$.

First, we accumulate sensor data. We set the interval of data accumulation $\Delta t_{data}$, the number of sequences of one batch $N_{seq}$, and the batch size $N_{batch}$. We accumulate $\{c_1, c_2, f\}^{data}$ which is $N_{seq}$ consecutive data of $\{c_1, c_2, f\}$ at intervals of $\Delta t_{data}$. Because $c_1$ cannot be directly obtained, we accumulate its estimated value explained in Section II-C. We accumulate the batch of $N_{seq}$ number of $\{c_1, c_2, f\}^{data}$, and begin to update the thermal model when the number of the batches exceeds $N_{batch}$. After finishing the update of the thermal model using $N_{batch}$ number of data batches, we remove the first of $N_{batch}$ batches, and update the model when the number of batches exceeds $N_{batch}$ again. These procedures are repeated online.

Next, we will explain the details of the online model update. Although we use only Eq. 9 for the estimation of $c_1$, we can also estimate $c_2$ by Eq. 10 at the same time. By comparing the estimated $c_2$ and the actual obtained $c_2$, we can update $P_{\{1,2,3,4,5\}}$. Among the accumulated data sequence of $\{c_1, c_2, f\}^{data}_{[k,k+N_{seq}-1]}$, first we pick out $c^{data}_{1,k}$, $c^{data}_{2,k}$, and $f^{data}_{[k,k+N_{seq}-1]}$. As $c_{1,k} = c^{data}_{1,k}$, $c_{2,k} = c^{data}_{2,k}$, and $f_{[k,k+N_{seq}-1]} = f^{data}_{[k,k+N_{seq}-1]}$, we repeat the equation below $N_{seq} - 1$ times.

$$c_{1,k+1} = c_{1,k} + h_1(f_k, c_{1,k}, c_{2,k}) \Delta t_{data} \tag{12}$$

$$c_{2,k+1} = c_{2,k} + h_2(c_{1,k}, c_{2,k}) \Delta t_{data} \tag{13}$$

Then, we can obtain $c_{1,[k+1,k+N_{seq}-1]}$ and $c_{2,[k+1,k+N_{seq}-1]}$. We compare the accumulated data of $c^{data}_{2,[k+1,k+N_{seq}-1]}$ and the data of $c_{2,[k+1,k+N_{seq}-1]}$ estimated using the current parameters, by the equation below,

$$L_{update} = \mathrm{MSE}(c_{2,[k+1,k+N_{seq}-1]}, c^{data}_{2,[k+1,k+N_{seq}-1]}) \tag{14}$$

where MSE represents mean squared error. Then, we update $P_{\{1,2,3,4,5\}}$ from this loss function by backpropagation through time [11], as shown below,

$$P_{\{1,2,3,4,5\}} \leftarrow P_{\{1,2,3,4,5\}} - \alpha \frac{\partial L_{update}}{\partial P_{\{1,2,3,4,5\}}} \tag{15}$$

where $\alpha$ is a learning rate. This is equivalent to updating weights of a neural network that represent parameters of the thermal model by a gradient descent method. Here, these parameters are actually updated using the average of the gradient calculated from the $N_{batch}$ data. Also, we set the maximum norm of the gradient as $D_{clip}$ by a gradient clipping method [12].

The problem of this method is that $c^{data}_{1,k}$ is not the actual sensor data but the estimated value. However, the change in $c_2$ is not very sensitive to the initial value of $c_1$, and from the subsequent experiments, we can verify that the parameters can be updated correctly.

In this study, we set $\Delta t_{data} = 1.0$ [sec], $N_{seq} = 30$, $N_{batch} = 10$, $\alpha = 0.02$, and $D_{clip} = 5.0$. These parameters are set from preliminary experiments to quickly converge the online update and not to exceed the time limit.

### E. Anomaly Detection of Motors

In this study, because the updated $P_{\{1,2,3,4,5\}}$ have physical meanings, we can conduct anomaly detection using the change in these parameters. This is the difference from a neural network whose parameters are difficult to interpret. We regard the current situation as anomaly simply when the value $g$ shown below exceeds a threshold $D_{detect}$,

$$g = \mathrm{RMSE}([P_1, P_2, P_3, P_4]^T, [P_1^{init}, P_2^{init}, P_3^{init}, P_4^{init}]^T) \tag{16}$$

where $P^{init}_{\{1,2,3,4\}}$ is $P_{\{1,2,3,4\}}$ at the start of the experiment, and RMSE represents root mean squared error. Because $P_5$ is a parameter of the ambient temperature and can always change, we do not use it for $g$. Also, when executing online learning for the first time, we start with $P_{\{1,2,3,4\}} = 0$, and

$P_{\{1,2,3,4\}}$ can change greatly. Therefore, anomaly detection should be executed after the parameters are firmly updated once. Also, one of the simplest methods of anomaly detection is displaying and monitoring these four parameters at all times.

In this study, we set $D_{detect} = 1.0$.

### F. Thermal Controller: Control of Motor Core Temperature

The thermal controller is a control to calculate the smooth sequence of maximum muscle tension $f^{limit}$ to rapidly achieve the maximum motor core temperature $c_1^{max}$ and to restrict motor output by this value. In this section, we will only explain the optimization part. We set the time interval $\Delta t_{control}$ and the number of sequences to consider $N_{control}$. Then, we optimize the sequence of $f$ during $N_{control}\Delta t_{control}$ seconds to achieve $c_1^{max}$ as rapidly as possible.

First, we represent the current estimated $c_1$ as $c_{1,t}^{current}$, and $c_2$ obtained from the thermal sensor as $c_{2,t}^{current}$. Also, we determine the sequence of $f$ before optimization $f^{limit}_{[k,k+N_{control}-1]}$. By setting $c_{1,k} = c_{1,k}^{current}$, $c_{2,k} = c_{2,k}^{current}$, and $f_{[k,k+N_{control}-1]} = f^{limit}_{[k,k+N_{control}-1]}$, we repeat the recurrence relations below $N_{control}-1$ times.

$$c_{1,k+1} = c_{1,k} + h_1(f_k, c_{1,k}, c_{2,k})\Delta t_{control} \tag{17}$$

$$c_{2,k+1} = c_{2,k} + h_2(c_{1,k}, c_{2,k})\Delta t_{control} \tag{18}$$

Then, we can obtain $c_{1,[k+1,k+N_{control}-1]}$ and $c_{2,[k+1,k+N_{control}-1]}$. We calculate the loss between $c^{max}_{1,[k+1,k+N_{control}-1]}$, which is a vector in which $N_{control}$ numbers of $c_1^{max}$ are arranged, and the estimated $c_{1,[k+1,k+N_{control}-1]}$ as below,

$$\begin{aligned} L_{control} = \mathrm{MSE}(c_{1,[k+1,k+N_{control}-1]}, c^{max}_{1,[k+1,k+N_{control}-1]}) \\ + W_{control}\mathrm{MSE}(\mathbf{0}, f^{limit}_{[k,k+N_{control}-1]}) \end{aligned} \tag{19}$$

where $W_{control}$ is a constant weight. By adding the minimization term of $f^{limit}$, the transition of $f^{limit}$ optimized from $L_{control}$ becomes smooth, and the stability of optimization increases. Although the term minimizing the difference of $f^{limit}$ at adjacent time steps is suitable for the purpose of this study, we found that the term destabilizes the optimization from preliminary experiments. By using the loss of $L_{control}$, $f^{limit}_{[k,k+N_{control}-1]}$ is updated by backpropagation through time [11], as shown below,

$$f^{limit}_{[k,k+N_{control}-1]} \leftarrow f^{limit}_{[k,k+N_{control}-1]} - \beta \frac{\partial L_{control}}{\partial f^{limit}_{[k,k+N_{control}-1]}} \tag{20}$$

where $\beta$ is a learning rate.

In actuality, we determine the maximum and minimum value of muscle tension $f^{\{min,max\}}$ for safety, and restrict $f^{limit}$. Also, the initial value of $f^{limit}_{[k,k+N_{control}-1]}$ is determined as $[f^{limit,T}_{[k,k+N_{control}-2]}, f^{limit}_{k+N_{control}-2}]^T$ by combining the shifted value of the previously optimized sequence of $f^{limit}_{[k-1,k+N_{control}-2]}$ and its last term. When it is the first time to optimize and there is no previously optimized value, we use a value in which is a vector that $N_{control}$ numbers of $f^{max}$ are arranged.

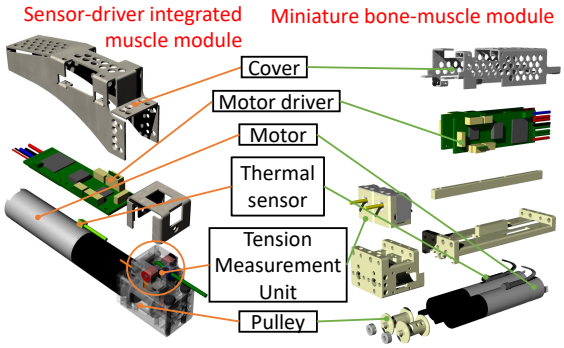


Fig. 4: Muscle actuator configuration in Musashi: sensor-driver integrated muscle module [14] and miniature bone-muscle module [15].

In this study, we set $\Delta t_{control} = 1.0$ [sec], $c_1^{max} = 80$ [°C], $N_{control} = 30$, $W_{control} = 0.001$, $\beta = 30$ [N], $f^{min} = 10$ [N], and $f^{max} = 300$ [N]. From preliminary experiments, $\Delta t_{control}$ and $N_{control}$ are set not to exceed the time limit, $c_1^{max}$ and $f^{\{min,max\}}$ are set within the range of ordinary use, and $W_{control}$ and $\beta$ are set to quickly stabilize the calculation of $f^{limit}$.

### G. Muscle Tension Limiter

The muscle tension limiter restricts $f$ by the maximum value of $f_k^{limit}$ calculated by Section II-F. Although we can set the maximum muscle tension easily when using tension-based control [13], when using length-based control, we elongate the muscle length and send $l^{ref} + \Delta l_k$ to the robot as below, as the muscle does not vibrate,

$$\begin{aligned} &if \;\; f > f_k^{limit} \\ &\qquad \Delta l_k = \Delta l_{k-1} + min(D_{gain}d - \Delta l_{k-1}, \Delta l_{plus}d) \end{aligned} \tag{21}$$

$$\begin{aligned} &else \\ &\qquad \Delta l_k = \Delta l_{k-1} + max(0 - \Delta l_{k-1}, -\Delta l_{minus}d) \end{aligned} \tag{22}$$

$$d = |f - f_k^{limit}| \tag{23}$$

where $|\cdot|$ is an absolute value, $\Delta l_{\{minus,plus\}}$ is a coefficient to determine the change in muscle length at one time step in the negative or positive direction, and $D_{gain}$ is a coefficient to determine the maximum elongation value. Thus, while restricting the change in muscle length by $\Delta l_{minus}d$ and $\Delta l_{plus}d$, muscle length is elongated or pulled as $f$ does not exceed $f_k^{limit}$. By using this control, even when an excessive muscle length is sent to the robot, the muscle actuator elongates automatically so that $c_1$ does not exceed $c_1^{max}$.

In this study, we set $\Delta l_{minus} = 0.001$, $\Delta l_{plus} = 0.003$, and $D_{gain} = 2.0$, and this control is executed every 8 msec. These parameters are set from preliminary experiments to limit muscle tension quickly and to not vibrate.

## III. Experiments

We conducted (A) a simulation and (B) actual motor experiments using one muscle actuator, as well as (C) using multiple muscle actuators of the musculoskeletal humanoid.

We show the configuration of muscle actuators of Musashi [9] used in this study: sensor-driver integrated muscle module

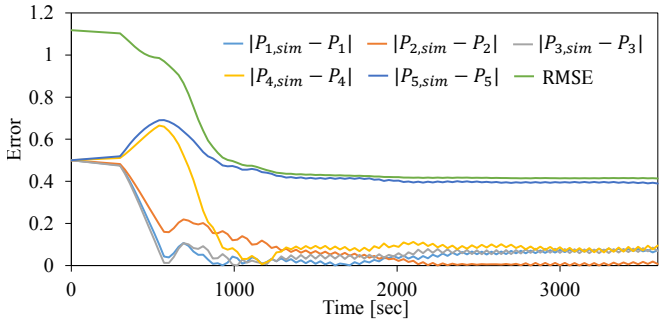


Fig. 5: Transition of errors in thermal model parameters between simulator and current model during online learning.

[14] and miniature bone-muscle module [15], in Fig. 4. Maxon brushless DC motor of EC-4pole 22 90W with 29:1 gear ratio is used for the former and EC 16 60W with 128:1 gear ratio is used for the latter. The former is used for the shoulder and elbow, and the latter is used for the wrist and fingers. As stated so far, a thermal sensor is attached to the motor housing. The wire of the muscle is wound by a pulley, and it comes out from the tension measurement unit, which can measure $f$. A motor driver, which can do current control, is attached to the motor, with a cover.

We describe the default parameters of these muscle actuators. Regarding EC-4pole 22 90W with 29:1 (sensor-driver integrated muscle module), $C_1 = 2.10$ [J/K], $C_2 = 29.0$ [J/K], $R_1 = 1.20$ [K/W], $R_2 = 10.3$ [K/W], and $K = 2.97E-4$ [J/N$^2$s]. Regarding EC 16 60W with 128:1 (miniature bone-muscle module), $C_1 = 1.19$ [J/K], $C_2 = 12.2$ [J/K], $R_1 = 4.30$ [K/W], $R_2 = 39.5$ [K/W], and $K = 4.50E-5$ [J/N$^2$s]. All parameters can be obtained from the datasheet of motors. Only $C_{\{1,2\}}$ cannot be directly obtained, but these parameters are approximated by $T_{\{1,2\}}/R_{\{1,2\}}$ ($T_{\{1,2\}}$ are thermal time constants of motor core and housing). Also, we set $c_a = 30$ [°C] as a default value.

### A. Simulation Experiment of One Muscle Actuator

*1) Online Learning:* We used a simulator of EC-4pole 22 90W with modified parameters of $P_{1,sim} = 0.5, P_{2,sim} = 0.5, P_{3,sim} = -0.5, P_{4,sim} = -0.5, P_{5,sim} = 0.5$. We set both the initial $c_1$ and $c_2$ as 30 °C. $f$ was updated by the equation of $f \leftarrow f + \mathrm{Rand}(-50.0, 50.0)$ every $\Delta t_{data}$ seconds ($\mathrm{Rand}(a,b)$ is a random value in $[a,b]$, and $f$ is limited within $[10,200]$ [N]), and was sent to the simulator. Only $c_2$ and $f$ can be observed from the simulator. There is a thermal model corresponding to the simulator, with default parameters of $P_{\{1,2,3,4,5\}} = 0$. By using this model, $c_1$ is estimated from the observed $c_2$ and $f$ (the initial value of $c_1$ is 30 °C), and online learning of the current model is executed using the obtained $(c_1, c_2, f)$. This procedure makes the parameters of the current model closer to those of the simulator. We show the absolute errors of parameters between the simulator and current model $|P_{\{1,2,3,4,5\},sim} - P_{\{1,2,3,4,5\}}|$ and the transition of RMSE of these five parameters over 3600 seconds in Fig. 5. We can see that each parameter of the current model became closer to that of the simulator, and the transition of errors converged in about 1200 seconds. Although the error of $P_5$ did not decrease to around 0, this is considered to be because the transition of $c_2$, which is effective in updating $P_5$, could not be obtained.

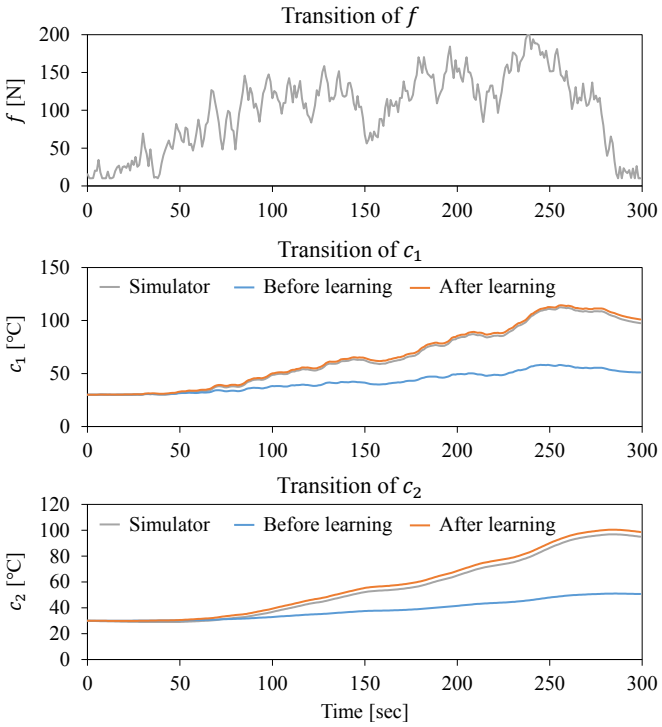


Fig. 6: Transition of $(f, c_1, c_2)$ of the simulator and the estimated $c_1$ and $c_2$ using the model before and after online learning.

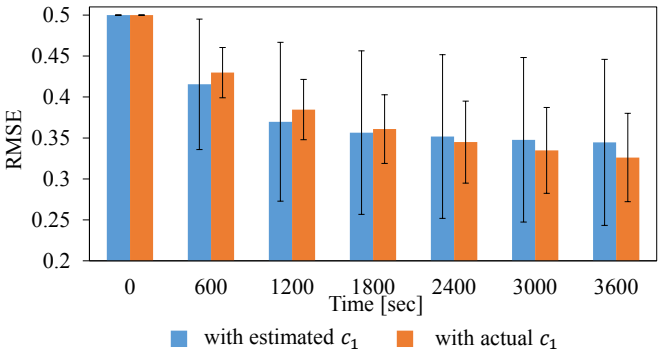


Fig. 7: Quantitative evaluation of online learning. The graph shows the transition of RMSE of thermal model parameters between the current model and the simulator with 10 modified parameters during online learning using the estimated or actual $c_1$.

Also, after executing online learning over 3600 seconds, we sent $f$ as stated above again to the simulator, the model before online learning, and the model after online learning. The result before online learning is the same with the result of the previous study [6]. We show the transition of $(f, c_1, c_2)$ in Fig. 6. The thermal estimation of $c_1$ and $c_2$ after online learning became closer to that of the simulator compared with before online learning. Although we used only $c_2$ as a teaching signal, $c_1$ also became closer to that of the simulator, and so this method is effective.

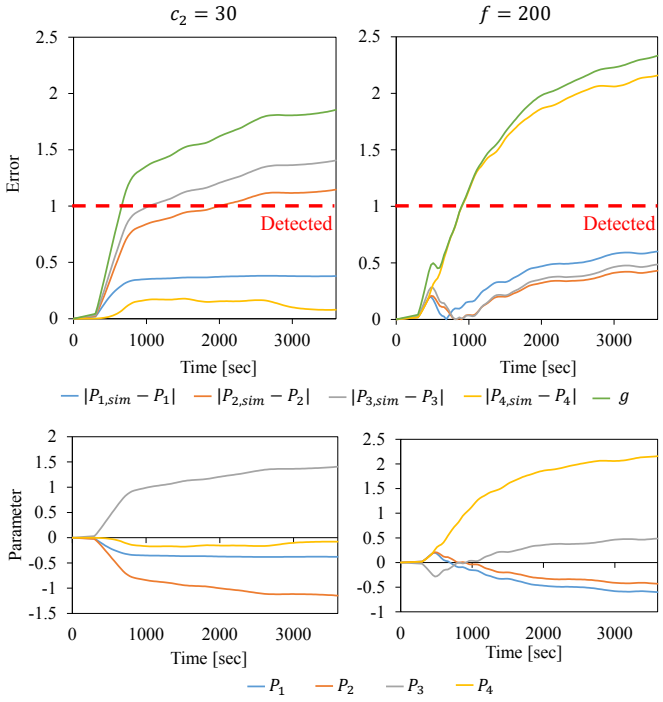


Fig. 8: Anomaly detection by learning thermal model parameters. The graph shows the transition of errors in thermal model parameters and $g$ during online learning.

*2) Quantitative Evaluation of Online Learning:* We constructed 10 parameters of simulators $P_{\{1,2,3,4,5\},sim}$ with additional random changes from the default parameters of $P_{\{1,2,3,4,5\}}$. Here, we set the RMSE between default and modified parameters to 0.5. We show the average and standard deviation of RMSE of five parameters between the current model and the simulator with 10 modified parameters every 10 minutes of online learning, when executing the same experiment with Section III-A.1, in Fig. 7. Here, we consider the problem explained in Section II-D, that not the actual but only the estimated $c_1$ can be obtained. Thus, we also show the difference of errors when using the actual and estimated $c_1$ as $c_1^{data}$ in Section II-D, in Fig. 7. As seen from the graph, when using the actual $c_1$ as $c_1^{data}$, compared with using the estimated one, the error decreased rapidly and its variance was small. On the other hand, although there was a certain degree of variance, the current model parameters also gradually became closer to those of the simulator when using the estimated $c_1$. Also, this result indicates that this study is applicable to various motors with different thermal parameters.

*3) Anomaly Detection:* We executed online learning by setting parameters of the simulator and current model to $P_{\{1,2,3,4,5\},sim} = 0$. Here, we consider two situations. One is a situation where the observed $c_2$ is always 30, because of the broken thermal sensor. Second is a situation where $f = 200$ is always sent to the simulator although the observed $f$ is correctly updated, because a gear or muscle wire is caught and the muscle does not move as intended. We show the transition of parameter errors $|P_{\{1,2,3,4\},sim} - P_{\{1,2,3,4\}}|$ and $g$ for anomaly detection during online learning for these

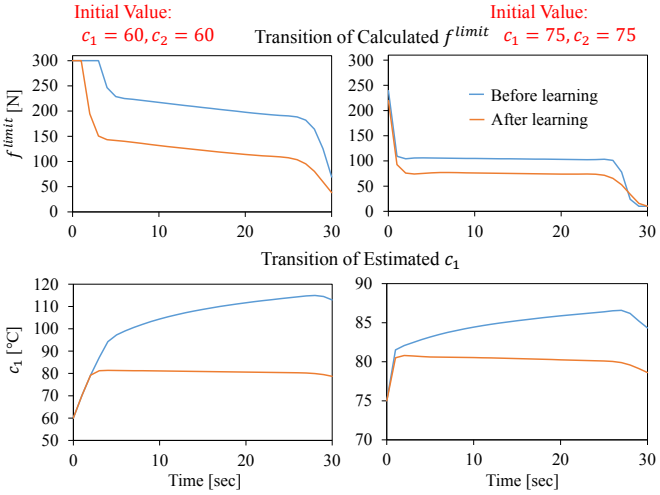


Fig. 9: Calculation of the optimized muscle tension to control $c_1$ and its evaluation. The graph shows the transition of the calculated $f^{limit}$ and observed $c_1$ with initial $c_1$ and $c_2$ of 60 or 75.

two situations, in the upper graph of Fig. 8. Also, in the lower graph of Fig. 8, we show the raw change of $P_{\{1,2,3,4\}}$. Regarding the former of the two situations, an anomaly was detected after 400 seconds of online learning, and regarding the latter, it was detected after 600 seconds. When looking at the values of $P_{\{1,2,3,4\}}$, regarding the former, the change in model parameters to constantly keep $c_2 = 30$ occurred. This was due to the increase in the heat escaping from the motor core to the motor housing by making $P_2$ small and decreasing heat capacity in motor core, and the decrease in the heat flowing in motor housing from the motor core by making $P_3$ large and increasing heat capacity in motor housing. Regarding the latter, by increasing $P_4$ and inhibiting the escape of heat from the motor housing to the ambient, a situation in which high $c_2$ is constantly observed due to $f = 200$ was produced. Thus, we can detect and interpret anomalies such as a broken thermal sensor, broken gear, and burnout of motors by using the interpretability of thermal model parameters.

*4) Thermal Controller:* Regarding the model of Section III-A.1 before and after online learning, when both the current $c_1$ and $c_2$ are 60 or 75, we observed the calculated $f^{limit}_{[k,k+N_{control}-1]}$. Also, we compared the transition of $c_1$ when sending the calculated $f^{limit}_{[k,k+N_{control}-1]}$ to the simulator, the model before online learning, and the model after online learning. As shown in Fig. 9, regarding both the initial temperatures of 60 and 75, the calculated $f^{limit}$ is high at first, gradually decreases, and is finally kept constant. Also, while $c_1$ achieved $c_1^{max}$ accurately when using the model after online learning, $c_1$ largely exceeded $c_1^{max}$ when using the model before online learning. Thus, online learning of the thermal model is effective for the thermal controller, and this method can always keep $c_1$ below $c_1^{max}$.

### *B. Actual Experiment with One Muscle Actuator*

*1) Online Learning:* We fixed the end position of the muscle, pulled $\boldsymbol{l}^{ref}$ by Rand$(-16, 0)$ [mm] over 10 seconds,

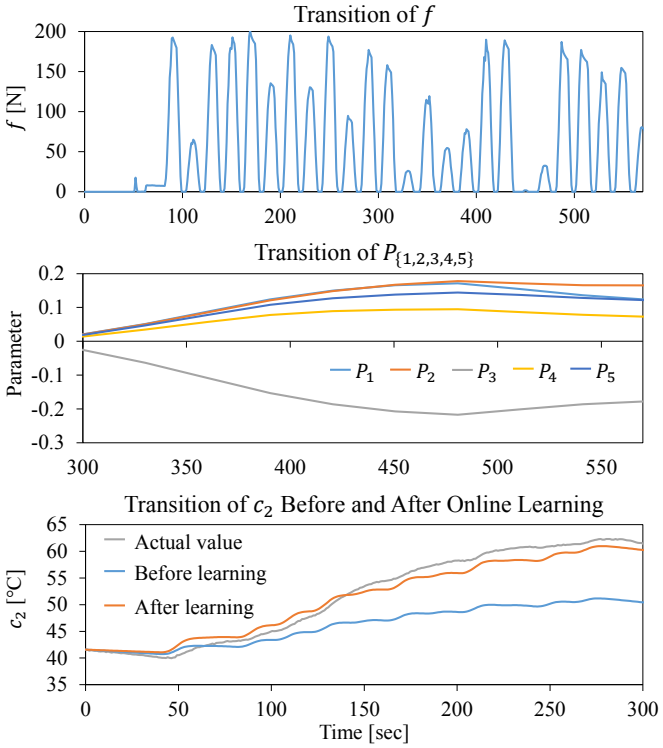


Fig. 10: Online learning experiment with one actual muscle actuator. The upper graph shows the transition of the applied $f$ during online learning, the middle graph shows the transition of $P_{\{1,2,3,4,5\}}$, and the lower graph shows the transition of the estimated $c_2$ before and after online learning.

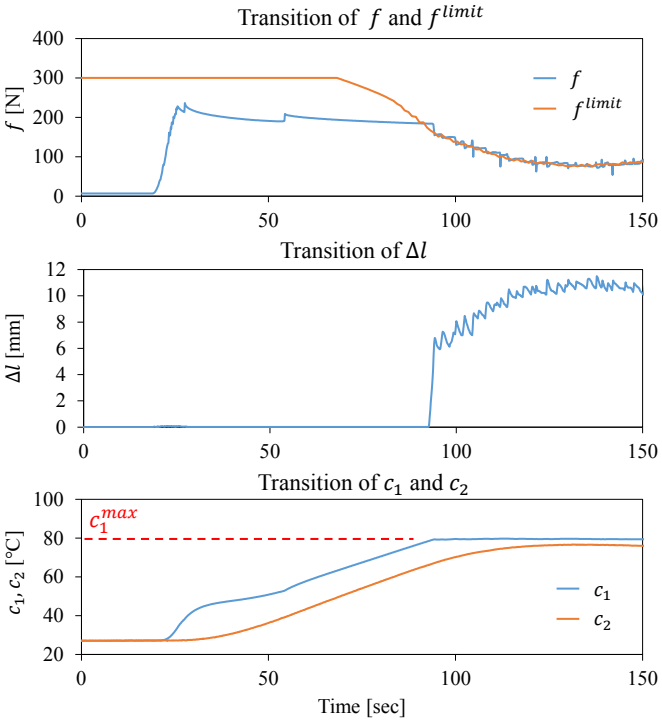


Fig. 11: Thermal control experiment with one actual muscle actuator. The upper graph shows the transition of $f$ and calculated $f^{limit}$, the middle graph shows the transition of $\Delta l$, and the lower graph shows the transition of $c_1$ and $c_2$.

elongated it over 10 seconds, and repeated these procedures. We show the transition of $f$ and $P_{\{1,2,3,4,5\}}$ during online learning in Fig. 10. After the online learning, when changing $\boldsymbol{l}^{ref}$ as stated above again, we compared the obtained $c_2$ from the actual muscle actuator and the estimated $c_2$ using the model before and after online learning, in the lower graph of Fig. 10. After online learning, the estimated value became closer to the actual sensor value than before online learning. Although the comparison of $c_1$ is difficult because the actual $c_1$ cannot be obtained, from Fig. 6 in Section III-A.1, the estimated $c_1$ is considered to become accurate by making $c_2$ correct. Therefore, this method is also applicable to the actual muscle actuator.

*2) Thermal Controller:* We verified whether $c_1$ can be always kept below $c_1^{max}$ by using the methods of Section II-F and Section II-G, and the updated model. We fixed the muscle endpoint, sent -16 mm for $\boldsymbol{l}^{ref}$, and generated $f$ of about 200 N. At the same time, by the muscle tension limiter of Section II-G, $f$ was restricted by $f^{limit}$. We show the transition of the observed $f$, calculated $f^{limit}$, $\Delta l$, estimated $c_1$, and observed $c_2$ in Fig. 11. $c_1$ gradually increased by keeping high $f$, and $f^{limit}$ began to decrease below $f^{max}$ in about 70 seconds. In about 90 seconds, $f$ exceeded $f^{limit}$, and $f$ was inhibited by gradually increasing $\Delta l$. We can see that $c_1$ constantly achieves $c_1^{max}$ by this control. Although $f^{max}$ should be about 100 N in order to keep $c_1$ below $c_1^{max}$, by using this study, large $f$ is ordinarily outputted but the output can be restricted by watching $c_1$.

### C. Application to the Musculoskeletal Humanoid

We consider how this study contributes to movements of the actual robot, by applying to the muscles of the left arm of Musashi [9]. While executing online learning, thermal estimation, thermal controller, and muscle tension limiter, we sent random joint angles to the robot as shown in Fig. 12. We moved the robot so that high $f$ is constantly generated by using variable stiffness control [16]. The important point here is not only limiting $c_1$ but also displaying the estimated $c_1$ with colors, as shown in Fig. 13. By monitoring $c_1$ updated accurately by online learning, we can rapidly find anomaly during experiments. Also, we show the transition of $f$ and $c_1$ of 10 muscles #1-#10 contributing to the shoulder and elbow of Musashi, in Fig. 14. The main muscles with high tension were #1, #2, #6, and #8. Although high $f$ of about 350 N occurred at the initial stage, as $c_1$ increased, the maximum $f$ was inhibited to about 250 N. Especially, the $c_1$ of #1, #2, and #6 vibrated around below 80, and we can see that the continuous movements of Musashi are achieved by the developed thermal controller. Therefore, this study is applicable to multiple muscle actuators, and is effective in movements of the actual robot.

In applying this study to the 36 muscles of both arms of Musashi, the calculation of the thermal estimator, online learning, and thermal controller takes 5, 200, and 200 msec, respectively. By rewriting the current program written in Python using C++ or modifying parameters of $N_{seq}$, $N_{control}$, and $N_{batch}$, this study can work at a higher frequency.

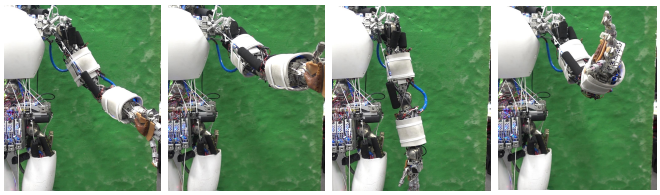

Fig. 12: Random movements of the left arm of the musculoskeletal humanoid Musashi [9].

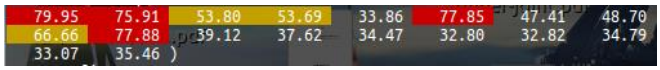


Fig. 13: Monitoring of $c_1$ with colors for manual thermal management. Each value shows $c_1$ included in the left arm of Musashi. When the value is colored red, the temperature is high (>70 °C), and when the value is colored yellow, the temperature is slightly high (>50 °C).

## IV. CONCLUSION

In this study, for smart management of motor temperature, we proposed an online learning method of thermal model parameters, estimation of motor core temperature, anomaly detection of motors, and control of motor core temperature by optimization. By referring to the observed motor housing temperature, the thermal model parameters can be updated online. Anomaly detection is easily enabled by using not a neural network but explicit parameters of the thermal model. The time series maximum output to achieve the maximum motor core temperature can be calculated by backpropagation, and motor core temperature can be managed by using it. Finally, we verified the effectiveness of this study in a simulation and in the actual robot, and achieved its application to the musculoskeletal humanoid.

In the current form, the thermal management can change the whole-body motion of the robot, and this can cause severe trouble for walking, task execution, etc. In future works, we would like to embed the calculated maximum output value in motion planning.

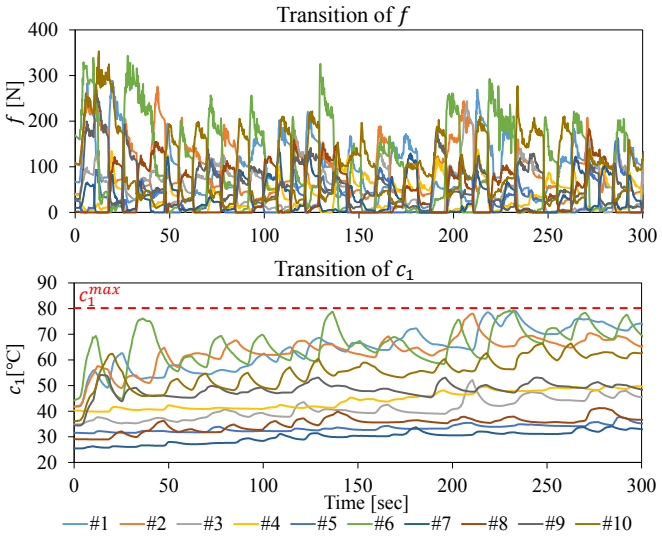


Fig. 14: Transition of $f$ and $c_1$ during random movements of the left arm of Musashi.

## REFERENCES


[1] J. Urata, Y. Nakanishi, K. Okada, and M. Inaba, "Design of high torque and high speed leg module for high power humanoid," in *Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 4497–4502.

[2] E. Sevinchan, I. Dincer, and H. Lang, "A review on thermal management methods for robots," *Applied Thermal Engineering*, vol. 140, pp. 799–813, 2018.

[3] T. Erez, K. Lowrey, Y. Tassa, V. Kumar, S. Kolev, and E. Todorov, "An integrated system for real-time model predictive control of humanoid robots," in *Proceedings of the 2013 IEEE-RAS International Conference on Humanoid Robots*, 2013, pp. 292–299.

[4] M. Diehl, H. G. Bock, H. Diedam, and P. B. Wieber, *Fast Direct Multiple Shooting Algorithms for Optimal Robot Control*. Springer Berlin Heidelberg, 2006, pp. 65–93.

[5] S. Noda, M. Murooka, S. Nozawa, Y. Kakiuchi, K. Okada, and M. Inaba, "Online maintaining behavior of high-load and unstable postures based on whole-body load balancing strategy with thermal prediction," in *Proceedings of the 2014 IEEE International Conference on Automation Science and Engineering*, 2014, pp. 1166–1171.

[6] J. Urata, T. Hirose, Y. Namiki, Y. Nakanishi, I. Mizuuchi, and M. Inaba, "Thermal control of electrical motors for high-power humanoid robots," in *Proceedings of the 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008, pp. 2047–2052.

[7] I. Kumagai, S. Noda, S. Nozawa, Y. Kakiuchi, K. Okada, and M. Inaba, "Whole body joint load reduction control for high-load tasks of humanoid robot through adapting joint torque limitation based on online joint temperature estimation," in *Proceedings of the 2014 IEEE-RAS International Conference on Humanoid Robots*, 2014, pp. 463–468.

[8] 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.

[9] 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.

[10] T. Kozuki, H. Toshinori, T. Shirai, S. Nakashima, Y. Asano, Y. Kakiuchi, K. Okada, and M. Inaba, "Skeletal structure with artificial perspiration for cooling by latent heat for musculoskeletal humanoid Kengoro," in *Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2016, pp. 2135–2140.

[11] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1*, D. E. Rumelhart, J. L. McClelland, and C. PDP Research Group, Eds. Cambridge, MA, USA: MIT Press, 1986, pp. 318–362.

[12] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *Proceedings of the 30th International Conference on Machine Learning*, 2013, pp. 1310–1318.

[13] M. Jäntsch, S. Wittmeier, K. Dalamagkidis, and A. Knoll, "Computed muscle control for an anthropomimetic elbow joint," in *Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 2192–2197.

[14] Y. Asano, T. Kozuki, S. Ookubo, K. Kawasaki, T. Shirai, K. Kimura, K. Okada, and M. Inaba, "A Sensor-driver Integrated Muscle Module with High-tension Measurability and Flexibility for Tendon-driven Robots," in *Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2015, pp. 5960–5965.

[15] K. Kawaharazuka, S. Makino, M. Kawamura, Y. Asano, Y. Kakiuchi, K. Okada, and M. Inaba, "Human Mimetic Forearm Design with Radioulnar Joint using Miniature Bone-muscle Modules and its Applications," in *Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2017, pp. 4956–4962.

[16] 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.