

PAPER

Learning and Control Model of the Arm for Loading

Kyoungsik KIM^{†,††a)}, *Student Member*, Hiroyuki KAMBARA^{††,†††}, *Member*, Duk SHIN^{††††}, *Nonmember*, and Yasuharu KOIKE^{††,†††}, *Member*

SUMMARY We propose a learning and control model of the arm for a loading task in which an object is loaded onto one hand with the other hand, in the sagittal plane. Postural control during object interactions provides important points to motor control theories in terms of how humans handle dynamics changes and use the information of prediction and sensory feedback. For the learning and control model, we coupled a feedback-error-learning scheme with an Actor-Critic method used as a feedback controller. To overcome sensory delays, a feedforward dynamics model (FDM) was used in the sensory feedback path. We tested the proposed model in simulation using a two-joint arm with six muscles, each with time delays in muscle force generation. By applying the proposed model to the loading task, we showed that motor commands started increasing, before an object was loaded on, to stabilize arm posture. We also found that the FDM contributes to the stabilization by predicting how the hand changes based on contexts of the object and efferent signals. For comparison with other computational models, we present the simulation results of a minimum-variance model.

key words: motor control, FDM, loading, actor-critic, feedback-error-learning

1. Introduction

Many studies have been done to find the mechanism that humans use to generate arm movements. Arm movements can be classified into two groups, postural control and reaching movements. Among the arm movements, mainly reaching movements have been studied, and the associated profiles, roughly straight hand paths with bell-shaped velocity, are well reconstructed by the assumption that the central nervous system (CNS) optimizes cost functions related to trajectory or muscle activations [1]–[3]. In the optimization models, predictions are mostly compared with horizontal reaching movements, which do not account for the influence of gravity. On the other hand, postural control has been studied relatively less, perhaps because postural control is considered comparatively easy or less important.

However, it has its own complexities, especially in environments under sudden dynamics changes in the sagittal plane. In such an environment, stable postural control will be difficult unless control problems (e.g. predictions of environmental changes, how to use visual and somatosensory information, delays in the sensory path) are correctly solved. Moreover, since there are infinitely numerous combinations of muscle activities for maintaining a position, postural control is an ill-posed problem. This is an important issue of motor control to be considered. In addition, arm movements in real life are affected by gravity and postural control occurs in many daily-life interactions, such as being handed a glass of water. Therefore, postural control in the sagittal plane is a valuable topic to study for generalizing motor control theories of the arm.

In a psychophysical experiment where a subject performs a loading task, loading an object onto one hand with the other hand, in the sagittal plane, it is confirmed that muscle forces are gradually increasing before the object contacts the hand [4]. This result indicates that muscle forces are adjusted in a feedforward way. It is, however, not clear how humans can learn and control motor commands in a feedforward way without prior dynamics knowledge, nor what the motor control model consists of.

One viable controller for realizing the feedforward way in biological systems is to use an inverse model [5]. The inverse model inverts the causal flow of the biological system, which generates from a given state of the arm motor commands that produced that state. Is solely an inverse model enough as a controller of the arm? Humans are able to respond in environments with sudden dynamics changes, such as a loading task or catching a ball, despite time delays of about 50-100 ms in the somatosensory loop when transferring sensory signals from sensory receptors to effector organs (e.g. arms and legs). For this to be possible, we believe a motor control model needs a module to compensate for possible movement errors due to time delays. Errors would not be generated if the inverse model outputted accurate motor commands for all environmental changes. It is, however, quite unlikely to have accurate inverse models for all cases in our daily lives. Therefore, the motor control model needs a feedback controller. In addition, a feedforward dynamics model (FDM) would be used in the feedback loop for state predictions and acquiring context information from the environment [6]. The usage of the FDM in the brain when interacting with objects is represented by Grip-force-load-force

Manuscript received June 17, 2008.

Manuscript revised November 20, 2008.

[†]The author is with the Interdisciplinary Graduate School Science and Engineering, Tokyo Institute of Technology, Yokohama-shi, 226–8503 Japan.

^{††}The authors are with Core Research for Evolution Science and Technology (CREST), JST, Kawaguchi-shi, 332–0012 Japan.

^{†††}The authors are with the Precision and Intelligence Laboratory, Tokyo Institute of Technology, Yokohama-shi, 226–8503 Japan.

^{††††}The author is with the TOYOTA CENTRAL R&D Laboratories INC., Aichi-ken, 480–1192 Japan.

a) E-mail: kyoungsik@hi.pi.titech.ac.jp

DOI: 10.1587/transinf.E92.D.705

coupling [7], [8]. It is also thought that the FDM minimizes the effects of sensory delays [9], [10].

In this study, we proposed a motor control model for loading based on the aforementioned principles. In the proposed model, an inverse statics model (ISM) is used as the inverse model, which handles the static component of the inverse dynamics. Therefore, when a desired state (e.g. joint angles) is given, the ISM outputs motor commands that make the arm converge to the corresponding equilibrium position. A feedback-error-learning scheme [11] is applied to train the ISM. Although Direct Inverse Modeling [12] and Forward and Inverse Modeling [9] can be used to train the inverse model, feedback-error-learning has more advantages because it acquires the inverse model while controlling biological systems that have redundancy. Alternatively, the Actor-Critic method which is one of the major frameworks for temporal difference (TD) learning in the Reinforcement Learning [13] is adapted to train the feedback controller in feedback-error-learning. Since it is difficult to imagine that we are born with a trained feedback controller, learning should be performed in a trial-error manner to acquire it [14], [15]. Finally, a FDM is used in the feedback path to overcome sensory delays. The validity of the proposed framework was already shown in the previous study, [14] for reaching movements.

The objective of this study was to determine how humans control the arm in an environment where sudden dynamics changes occur (e.g. loading) and how to use the information of prediction and sensory feedback. Calling the proposed model a learning and control model, we use the term “learning” to convey that the proposed model can acquire control laws without prior dynamics knowledge. In the simulation of the proposed model, a loading task was performed. Hand trajectories, motor commands and hand force were represented using a two-joint arm with six muscles that have delays in signal transfer. Results showed that motor commands gradually increased, before the object made contact with the hand to stabilize posture based on predicted information from the FDM. The simulation data showed tendencies similar to the psychophysical experimental data in the view of hand trajectories and hand force.

2. Proposed Model

2.1 Learning and Control Model

The learning and control model is shown in Fig. 1. In the simulation, the state space, $\theta = (\theta_S, \theta_E, \dot{\theta}_S, \dot{\theta}_E)$, consists of angles and angular velocities of the shoulder and elbow joint. However, angular velocities in the desired state space, $\theta^d = (\theta_S^d, \theta_E^d, 0, 0)$, are always set to 0 because the aim of the task is to maintain the arm at a desired position. Therefore, the ISM is used as an inverse model instead of an inverse dynamics model (IDM) which includes angular velocity and acceleration.

As shown in Fig. 1, the ISM receives the desired state θ^d and outputs feedforward motor commands u^{ff} for six

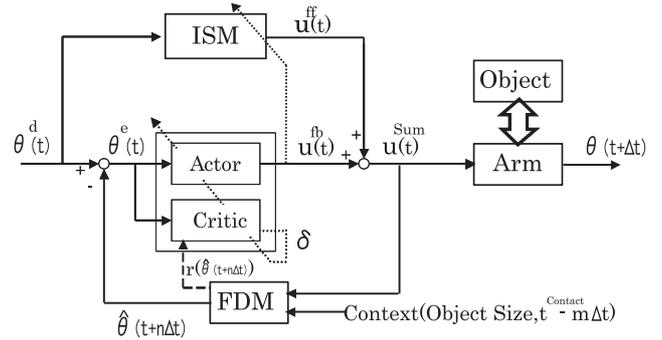


Fig. 1 Learning and control model using feedback-error-learning and Actor-Critic method. When a desired state, θ^d , is given, the forward dynamics model predicts the state of the arm at $n\Delta t$ ahead using context information and motor commands, in order to control the arm that has sensory delay. Then, the ISM outputs motor commands for the arm to hold the desired state. The actor outputs motor commands to modify predicted errors between the desired state and predicted state.

muscles. Meanwhile, Actor receives the state error θ^e between the desired state θ^d and predicted state $\hat{\theta}$, and produces feedback motor commands u^{fb} for six muscles. The sum of u^{ff} and u^{fb} is fed to the controlled object, “Arm”, as motor commands, u^{sum} . Then, the actual state θ is updated according to the dynamics of the controlled object and the environment. Meanwhile, motor commands and context signals are inputted to the FDM to generate the predicted state, $\hat{\theta}$.

Next, we calculate the reward r , a function of the predicted state that evaluates how close the predicted state is to desired state. Based on the reward and value function outputted from Critic, a reinforcement signal δ is calculated [16]. Improvements of Actor and Critic are carried out by δ . If δ is positive, it means that u^{sum} just taken worked better than expected. Thus, the tendency to select u^{sum} will be strengthened. Whereas if δ is negative, then the tendency to select u^{sum} will be weakened. Meanwhile, the output of Actor u^{fb} is used as the error signal for improving the ISM. Then, the ISM learned to make the u^{fb} zero.

2.2 Implementations of Actor-Critic, ISM and FDM

2.2.1 Actor-Critic Method

We used TD (λ) learning and a continuous Actor-Critic method for the implementations of Actor-Critic [16] since the treated spaces and time are continuous. Normalized gaussian network (NGNet) [16] were used as a function approximator for Actor-Critic. The inputs to Actor and Critic are both $\theta(t)^e$. We denote this $\theta(t)^e$ simply as $x(t)$ for compactness of notation ($x(t) \doteq \theta(t)^e$).

In the Actor-Critic method, Critic learns so that the value function $V(x(t))$ for input state, $x(t)$, is estimated correctly using reward values and outputs estimated evaluation value that presents how desirable the input state is. Meanwhile, Actor learns optimal control law $\mu(x(t))$ and selects an action $u^{actor}(t)$ based on the law so as to move $x(t)$ to a

higher evaluation. The estimate of value function is

$$V(\mathbf{x}(t)) = \sum_{k=1}^N w_k^C b_k(\mathbf{x}(t)) \quad (1)$$

and the i -th output of Actor, $u_i^{actor}(t)$ is determined as

$$u_i^{actor}(t) = g\left(\sum_{k=1}^N w_{i,k}^A b_k(\mathbf{x}(t)) + \sigma(t)n_i(t)\right) - 0.5 \quad (2)$$

where $b_k(\mathbf{x}(t))$ is a basis function of NGNet [16], N is the number of basis functions. w_k^C and $w_{i,k}^A$ is the weights for the Critic and Actor networks. The function $g()$ is the sigmoid function to saturate the output. $n_i(t)$ is the white noise for exploration to find a better motor command, and σ is used to determine the magnitude of noise. σ is defined as

$$\sigma(t) = |\hat{\delta}| \quad (3)$$

where $|\hat{\delta}|$ is a value that takes the average of absolute $\delta(t)$ for the previous trial (see Sect. 3.3 for the explanation of the trial).

TD error is calculated for the learning of Actor-Critic as follows

$$\delta(t) = r(t) - \frac{1}{\tau}V(\mathbf{x}(t)) + \dot{V}(\mathbf{x}(t)) \quad (4)$$

where $r(t)$ is a reward and τ is a decay time constant of the value function [16]. The weights in Critic are updated by

$$w_k^C = \beta^C \delta(t) e_k^C(t) \quad (5)$$

where β^C is the learning rate and $e_k^C(t)$ is an eligibility trace that presents how much effect the k th basis function has had on Critic's outputs up to time t [13]. The eligibility trace is exponentially weighted as

$$e_k^C(t) = \int_0^t \exp\left(-\frac{t-s}{\kappa}\right) b_k(s) ds \quad (6)$$

where κ is a decay time constant. At the same time, the weights in Actor are updated as

$$w_{i,k}^A = \beta^A \delta(t) e_{i,k}^A(t) \quad (7)$$

where β^A is the learning rate and $e_{i,k}^A(t)$ is the eligibility trace. The equation of eligibility trace is as follows

$$\begin{aligned} e_{i,k}^A(t) &= \int_0^t h(t-s) d_{i,k}(s) ds \\ d_{i,k}^A(s) &= \sigma(s) n_i(s) b_k(\mathbf{x}(t)) \end{aligned} \quad (8)$$

Here fuction $h(z)$ is the impulse response of a second order low pass filter applied to the motor commands \mathbf{u}^{sum} to simulate sensory delays [17]. The eligibility trace of Actor is weighted with the shape of the filter. The impulse response function is defined as

$$h(z) = \frac{bc}{c-b} (\exp(-bz) - \exp(-cz)) \quad (9)$$

where b is 10.80 and c is 16.52. The function is normalized so that its size becomes 1.

2.2.2 Inverse Statics Model (ISM)

The network of the ISM is as follows

$$u_i^{ISM}(t) = g\left(\sum_{k=1}^N q_{i,k} b_k^{ISM}(\boldsymbol{\theta}^d(t))\right) \quad (10)$$

$b_k^{ISM}(\boldsymbol{\theta}^d(t))$ is a normalized gaussian basis function [16]. N is the number of the basis functions. $q_{i,k}$ is a weight from the k th basis function to the i th output. $g()$ is a sigmoid function. The update rule of the ISM is

$$\dot{q}_{i,k} = \beta^{ISM} \tilde{u}_i^{actor}(t) b_k^{ISM}(\boldsymbol{\theta}^d(t)) \quad (11)$$

where $\tilde{u}_i^{actor}(t)$ is the lowpass filtered Actor's output.

$$\tilde{u}_i^{actor}(t) = \int_0^t h(t-s) u_i^{actor}(s) ds \quad (12)$$

β^{ISM} is a learning coefficient that changes depending on TD error. The value of β^{ISM} increases with decreased TD error and is expressed as follows

$$\beta^{ISM} = \beta_0 \exp\left(-\frac{|\hat{\delta}|^2}{s_{lr}^2}\right) \quad (13)$$

$|\hat{\delta}|$ is a value that takes the average of absolute $\delta(t)$ for the previous trial. β_0 and s_{lr} are constants. Note that this error signal \mathbf{u}^{fb} does not work correctly until the improvement of Actor reaches some level. For this reason, we make the learning rate of the ISM increase as TD error, δ , decreases. This idea comes from the fact that the magnitude of the TD error decreases as the improvement of Actor and Critic advances. It is one of the properties in reinforcement learning [13].

2.2.3 Forward Dynamics Model (FDM)

FDM predicts states of the arm at time $n\Delta t$ later based on current motor commands, \mathbf{u}^{sum} , and the size of the object as context information. When interacting with an object, one needs to estimate the weight of the object to consider how the object affects control. The weight can be estimated from material and size of the object. Here, we simply decide the weight using a linear relationship between the size and the weight as *object weight* [g] = 100**object size* (e.g. if the object size is 0.6, then the weight is decided as 600 g). Another parameter used in the FDM is the timing of that the object is considered. Even if one knows the existence of the object from visual information at the beginning of the task, the time when the object should be considered in the control mechanism must be decided. Here, we assumed that the context information of the object is considered in the mechanism and inputted to the FDM from $t = t^{contact} - m\Delta t$ where $t^{contact}$ is the time at which the object contacts the hand.

Meanwhile, the implementation of the FDM is done by

using a dynamics equation (see Sect. 3.1, Sect. 3.2 and Appendix A). In fact, the FDM should be trained in a real biological system. However, for simplicity of learning, dynamics equations are used to implement the FDM in this simulation. Note that the FDM can be acquired while performing the task using a neural network [14]. Furthermore, it was found through psychophysical experiment that the FDM predicting the dynamics of an object and arm was acquired before a control model (e.g. inverse model) is trained [18], the assumption of using a trained FDM is not thought to be a discrepancy.

3. Simulation

3.1 Arm Model

We used the two-link arm with six muscles as shown in Fig. 2. At first, to express sensory delays in muscles that occur when generating muscle force from reached motor commands, a low-pass filter is applied to the motor commands [14], [17]. In neurophysiological studies, it is known that muscle forces can be well predicted by low-pass-filtering the neural impulses with a second-order filter [19]. Then, the filtered motor commands are transferred to the muscle model to calculate muscle tensions. The muscle model can be written in Eq. (14) expressing the muscle as an elastic element and a viscous element arranged in parallel [20]. Each muscle is designed to be a nonlinear actuator, in which muscle tension, T , depends on the motor commands, length and contraction velocity of the muscle. T is defined as

$$T_i = (k_0^{(i)} + k_1^{(i)} \tilde{u}_i^{sum}) \left(l_0^{(i)} + l_1^{(i)} \tilde{u}_i^{sum} - \sum_{j=1}^2 a_{ij} \theta_j \right) - (b_0^{(i)} + b_1^{(i)} \tilde{u}_i^{sum}) \left(\sum_{j=1}^2 a_{ij} \dot{\theta}_j \right) \quad [i=1, 2, \dots, 6.] \quad (14)$$

Muscle
sh flx : shoulder flexor
sh ext : shoulder extensor
el flx : elbow flexor
el ext : elbow extensor
dj flx : double joint flexor
dj ext : double joint extensor

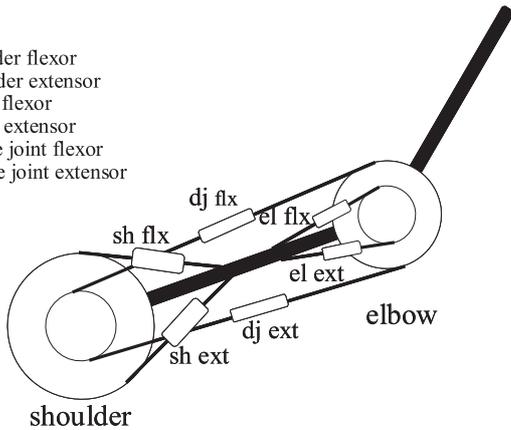


Fig. 2 Arm model: two-link arm with six muscles is used. The upper arm is moved by one pair of biarticular muscles and one pair of monoarticular muscles around the shoulder. The forearm are moved by one pair of biarticular muscles and one pair of monoarticular muscles around the elbow.

Here, \tilde{u}_i^{sum} is a filtered motor command of u_i^{sum} to i -th muscle. k , b and l are the elasticity, viscosity and muscle length coefficients, respectively. i is an index of muscles. j is an index of joints. Since the controlled object is a two-link arm with six muscles, the values of i and j are 6 and 2, respectively. a is the moment arm. θ_j is the joint angle. $\dot{\theta}_j$ is the joint angular velocity. All coefficients are determined and adjusted properly for the simulation task, based on data from [20] and [21]. Coefficient values are shown in Appendix A. Joint torque, τ , is calculated as

$$\tau_j = \sum_{i=1}^6 a_{ij} T_i \quad [j=1, 2.] \quad (15)$$

Joint torque, τ , was calculated using Eq. (15). The j and i are indexes of joints and muscles, respectively. Therefore, once the motor commands are determined from the outputs of the ISM and Actor, muscle tensions and torques can be calculated from Eq. (14) and Eq. (15).

3.2 Object Model

The object is expressed by transforming the gravity force of the object acted on the hand to joint torque. For example, when holding an object in the hand at a certain position, the joint torque for the object is required in addition to the joint torque needed for the arm in order to maintain arm posture (see Appendix A for details). The joint torque of the object is

$$\begin{pmatrix} \tau_s^{obj} \\ \tau_e^{obj} \end{pmatrix} = J^T \begin{pmatrix} F_x^{obj} \\ F_y^{obj} \end{pmatrix} \quad J = \begin{pmatrix} -l_1 \sin(\theta_s) - l_2 \sin(\theta_s + \theta_e) & -l_2 \sin(\theta_s + \theta_e) \\ l_1 \cos(\theta_s) + l_2 \cos(\theta_s + \theta_e) & l_2 \cos(\theta_s + \theta_e) \end{pmatrix} \quad (16)$$

where J^T is the transposed form of the jacobian matrix. τ_s^{obj} and τ_e^{obj} are the joint torque acting on the shoulder and elbow respectively by the object. F_x^{obj} and F_y^{obj} are the forces applied to the hand in the horizontal axis and the sagittal axis, respectively. Here, we assumed that the object force is applied only to the sagittal axis which is in the gravity direction. Thus, F_x^{obj} is set to 0.

3.3 Task: Postural Control during a Loading Task

At first, success in controlling the arm is needed in order to perform the loading task. Hence, after learning in the state where there is no object, the model is then trained for loading using the weight file acquired in the previous learning. In the simulation, one object is used, with the weight and size of the object are set to 600 g and 0.6, respectively. At the beginning of each trial, the desired state, θ^d , and initial state, θ , take the same value chosen randomly from the desired range (Sect. 3.5) and are fixed during the trial. During the first half interval, postural control is performed for the arm, and the control is performed for 1 [s] more after the object, 600 g, is loaded on at $t^{contact} = 0$ [s] (Fig. 3). This is

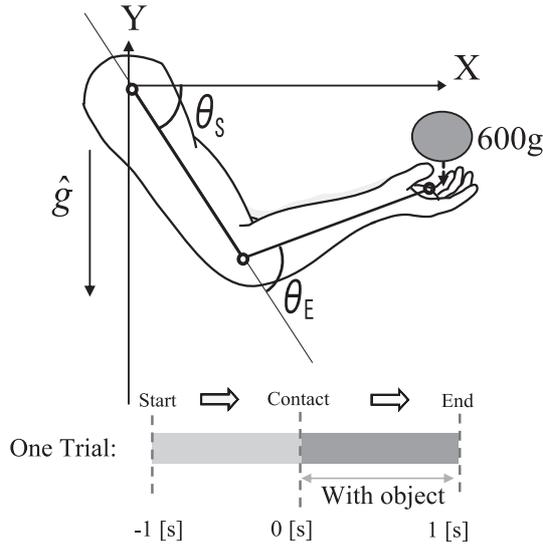


Fig. 3 Loading task: for a desired state (θ_s, θ_e) , postural control is performed for 2 seconds in the sagittal plane where gravity \hat{g} acts. During the trial, a 600 g object is loaded on the hand at 0 [s]. $\theta_s = 0$ is the state when the upper arm is laid in the horizontal axis. $\theta_e = 0$ is the state when the lower arm is in the axis extended from the upper arm. The angles take positive values in the direction of flexing.

one trial. It is assumed that the object loaded is reflected to the FDM at 80 ms ($m\Delta t$ in Fig. 1) before the object contacts the hand. The FDM predicts 80 ms ($n\Delta t$ in Fig. 1) later from current state. Note that when the state has exceeded a realizable range, θ^r (Sect. 3.5), the trial is terminated and training moves to the next trial.

3.4 Reward

Reinforcement learning is a learning method that acquires actions so as to get higher rewards. Therefore, it is important to set up reward functions that have the maximum value near the goal. In this task, our goal is to acquire motor commands which maintain the hand at a desired position. Thus, we determined a reward, $r(t)$ in Fig. 1, for arm posture as follows

$$r(t) = 2 \left(\exp \left(-\frac{d(t)^2}{s_r^2} \right) - 0.5 \right) \quad (17)$$

where $d(t)$ [m] is the distance error between desired hand position and predicted state. s_r is the constant which adjusts how severely the distance error is evaluated. Its value is set to 0.1. In the simulation, $r(t)$ gets more rewards as the predicted state approaches the desired position.

3.5 Simulation Conditions during Learning

The inputs for the model (Fig. 1) are the desired state, $\theta^d = (\theta_s^d, \theta_e^d, \dot{\theta}_s^d, \dot{\theta}_e^d)^T$, for the ISM and error states between the desired state and predicted state, $\theta^e = (\theta_s^e, \theta_e^e, \dot{\theta}_s^e, \dot{\theta}_e^e)^T$, for Actor. The low subscript S and E indicate the shoulder and elbow, respectively. The output ranges of the ISM and Actor are set to 0 ~ 1 and -0.5 ~ 0.5, respectively. Note that

outputs of the ISM take positive values for both flexors and extensors. In contrast, outputs of Actor, u^{fb} , include negative values, in order to extract excessive force in muscle. In addition, $u^{sum} = u^{ff} + u^{fb}$ is regulated from 0 to 1 for the normalization of motor commands by setting it to 0 when it takes a value below 0 and setting it to 1 when it takes a value over 1. The state range realized in simulation, θ^r , is $-120 \leq \theta_s \leq 0$ [deg], $0 \leq \theta_e \leq 130$ [deg], $-900 \leq \dot{\theta}_s, \dot{\theta}_e \leq 900$ [deg/s], $-900 \leq \dot{\theta}_s, \dot{\theta}_e \leq 900$ [deg/s]. Desired state, θ^d , is randomly chosen in the range of $-90 \leq \theta_s^d \leq -50$ [deg], $30 \leq \theta_e^d \leq 90$ [deg], $\dot{\theta}_s^d = \dot{\theta}_e^d = 0$ [deg/s] [22]. The time step is set to $\Delta t = 0.01$ seconds (Fig. 1). And renewal of motor command, future state prediction and arm's dynamics are all performed for every Δt . The networks' updates of the ISM and Actor-Critic are also performed for every Δt . Other parameters are set as follows: $\tau = 1.0, \kappa = 0.8, \beta^C = \beta^A = 0.3, N = 6561$ ($9 \times 9 \times 9 \times 9$ for angles and angular velocities of the shoulder and elbow) in Sect. 2.2.1. $\beta_0 = 0.08, s_{lr} = 0.1, N = 225$ (15×15 for angles only since the angular velocities are 0 for the loading task) in Sect. 2.2.2.

4. Simulation Results

The learning process is shown in Fig. 4. As the model learns, each parameter converges on a value, which gives a criterion whether the learning is sufficiently performed. In this figure, the reward value is high from the beginning because we used pre-learned weight files of postural control without the object. The model in the loading task, however, starts to learn in the direction of getting more reward when the task changes, showing that the model copes with instability caused by fast dynamics changes. Furthermore, motor commands in the later trials increase compared to early trials due to the bearing of the disturbance. Here, in order to compare simulation features with the psychophysical experiment [4], a weight file (30000th) sufficiently trained is chosen using a learning rate and reward as an index. With this weight file, simulation of loading is performed at the same desired state as the psychophysical experiment $(\theta_s, \theta_e) = (-50, 65)$. The results are shown in Fig. 5. As shown in the figures, the hand position is stable before and after the environment changes. This stable control is possible by the following process. First, the FDM predicts how the object influences the arm. Distance errors between the desired state and predicted state are calculated. Then the feedback controller, Actor, generates motor commands to decrease the predicted distance errors. The renewed motor commands transmitted to the arm, with sensory delays, make the motor commands increase slowly before fast dynamics change occurs. In the simulation, hand position changed about 3 cm in the vertical direction and the distance of the lowest fallen position from the desired position was about 1.6 cm (Fig. 5 (b)). In a psychophysical experiment of loading tasks for 4 different weights (400 g, 600 g, 800 g, 1000 g) [4], 4 subjects were asked to hold a weight with their left hand just above their right hand and load the object on the right hand at the sound of a beep signal. Each subject performed the loading task 20

times for each weight. During the experiment, hand force and hand position were measured using electromyography and an Optotrack (Northern Digital Inc., Canada). According to the experiment, there were approximately 2 cm distance errors in average between the desired position and the lowest fallen position for three out of four subjects. Furthermore, in the psychophysical experiment, it was observed that hand force arrived at its peak immediately after loading the object. In the simulation, it arrived at the peak about 250 ms after the contact. This indicates that a tendency similar to the psychophysical experiment is reproduced in the simulation using the FDM. In Fig. 6, the distance of the lowest fallen position from each desired position is shown. This result indicates that the loading task is possible for all desired positions within the desired range.

We tested the generalization capability of the model in real environments that humans may be confronted with when performing a loading task. Firstly, in a real environment, the weight prediction of an object by humans can be

wrong. For example, objects of equal weight and size are perceived differently due to the material (e.g. balsa wood, steel) [23]. In a size-weight illusion, a smaller object is perceived heavier than a larger object when objects of equal weight and different size are lifted by hand [24]. For the simulation of this situation, we performed loading tasks under the assumption that the weight prediction of an object in the FDM was not correct. We show the results in Fig. 7. In each loading task, the weight of the object loaded was fixed at 600 g, but the object size used as context information was changed from 0.4 to 0.8 at intervals of 0.1, resulting in the FDM predicting the weight from 400 g to 800 g. As shown in the figure, the hand fell lower when the weight prediction was lighter than 600 g and fell less when the weight prediction was heavier. A similar tendency is expected in real loading tasks for humans. Secondly, we performed loading tasks for unlearned object weights. In real environments, it

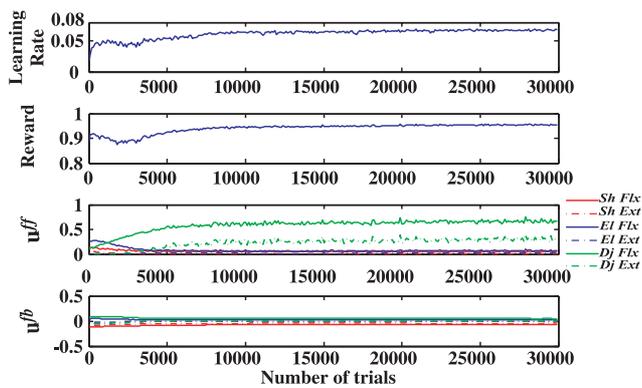


Fig. 4 Learning Process: learning rate of the ISM, reward, ISM output and actor output from the top in order. Data was averaged over 2 [s] of one trial then smoothed using a low pass filter with 100 trial width.

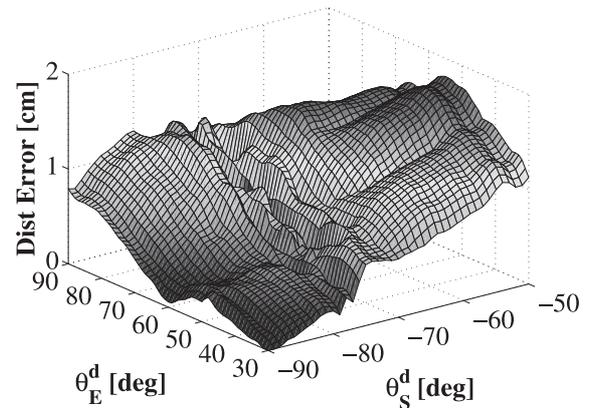


Fig. 6 Distance error between the lowest fallen position and each desired position in the vertical direction after loading: 2501 (41×61) desired positions on the equally divided grid of the desired range are tested. The average and standard deviation is 0.971 ± 0.433 [cm].

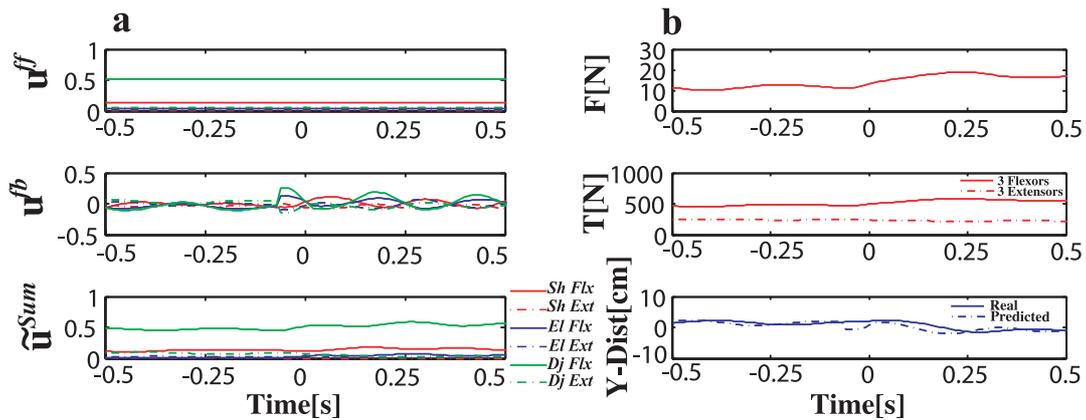


Fig. 5 Simulation results of the loading task for a desired state $(-50, 65)$ by using the 30000th weight file. (a) Motor commands of the ISM and actor and filtered motor commands are shown. Actor changes its output to modify predicted distance error before the object contacts the arm. (b) F [N]: Hand force in the vertical direction, T [N]: Summed tensions for three flexors and three extensors, Y -Dist [cm]: Hand position errors ($real:\theta(t + \Delta t)$, $predicted:\hat{\theta}(t + n\Delta t)$ in Fig. 1) in the vertical direction from the desired position. Before the object is loaded on at 0 [s], the FDM predicts that the hand will fall down.

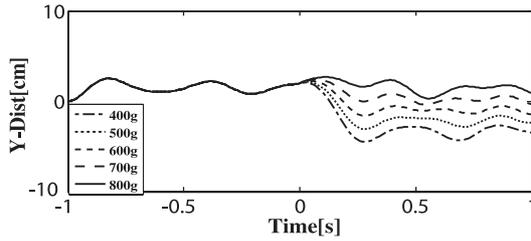


Fig. 7 Simulation results of the loading task when the predicted weight in the FDM is different from the weight loaded. The simulation was performed using the 30000th weight file for a desired state $(-50, 65)$. In each loading task, one of the 5 weight predictions from 400 g to 800 g was used, but the weight of the loaded object was fixed at 600 g.

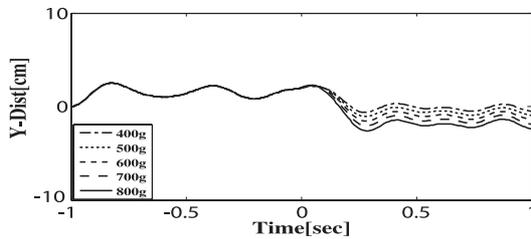


Fig. 8 Simulation results of the loading task for unlearned weights. The simulation was performed at a desired state $(-50, 65)$ using the 30000th weight file trained only for 600 g. In loading tasks for the 5 objects, it was assumed that the predicted weight from the FDM was the same as the weight of a loaded object.

is difficult to believe that one should always learn the weight of an object in order to perform a loading task for that object. For this reason, it is expected that an acquired control model for an object’s weight is utilized to perform loading tasks for objects of unlearned weights. For the simulation of this situation, we performed loading tasks, using the model trained at 600 g, with other weights under the assumption that FDM predictions were correct. The results are shown in Fig. 8. As shown in the figure, loading tasks are possible for unlearned weights (e.g. 400 g and 800 g). These two results indicate that the model can perform loading tasks in the situations of wrong weight predictions and unlearned weights, which are very likely to occur in real environments. These are likely the generalization capabilities that humans possess and are needed to interact with objects in our daily lives.

5. Discussion

5.1 Effects of the FDM

To test the effectiveness of the FDM, a simulation based on the delayed feedback signal without using predicted state (Fig. 9) was performed for the three delay times (Delay: 80 ms, 30 ms, 0 ms) in the same conditions described in Sect. 3.5. Its results are presented in Fig. 10. Here, the FDM is not used. Thus the reward is evaluated from the feedback signal since the predicted state is not available. Also note that as mentioned in Sect. 3.3, the trial is terminated when the hand has exceeded the realizable range, θ^r . From Fig. 10, the graph of trial time describes that in the case of

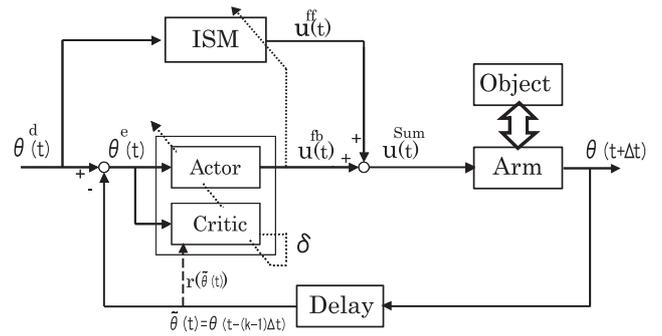


Fig. 9 Effects of the FDM: $k = 0, 3, 8$ are tested for 3 cases of delay times to see if stable control is possible without using the FDM.

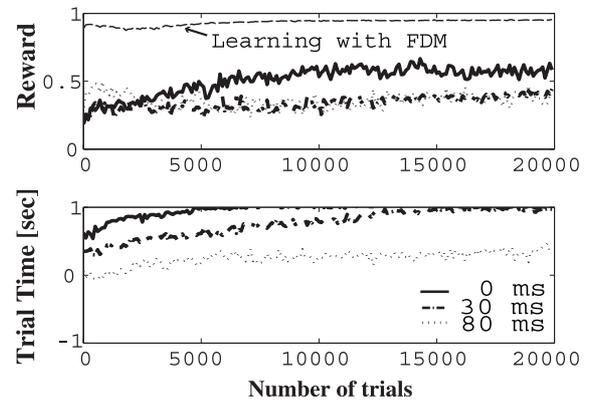


Fig. 10 Delayed feedback effects: The data shows the effects on the loading task when using delayed sensory feedback. In the upper graph, the reward graph in Fig. 4 (Learning with FDM) is presented again to aid the comparison of results between delayed sensory feedback and the predicted state. In the lower graph, trial time shows the duration that the hand stays in the realizable range, θ^r (see Sect. 3.5). All data have been filtered with 100 trial width for smoothing.

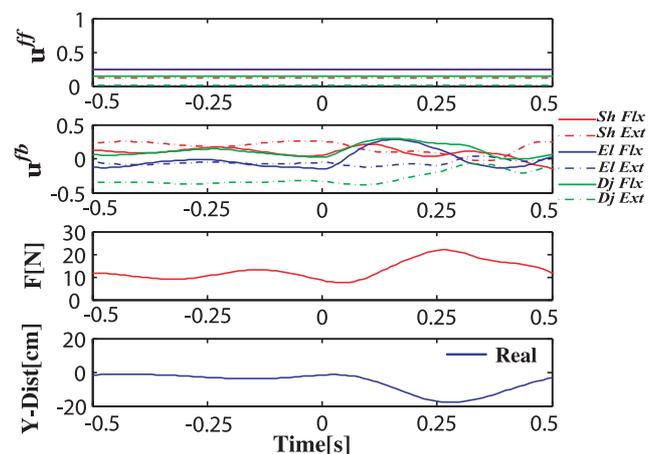


Fig. 11 Simulation results of the loading task for a desired state $(-50, 65)$ by using the 15000th weight file when the FDM is unused and delay time is 0 ms. Motor commands of the ISM and Actor, hand force and hand distance error are shown.

a long delay time, the hand position goes out of the range just after the hand contacted the object at 0[s]. However, the time interval for staying within the range gets longer

as the given delays shorten. This, however, does not mean that maintaining posture was successfully performed without a using prediction, even though the ability of control increased. Moreover, the reward values for all cases (Fig. 10) were smaller than in using the FDM (Fig. 4), which means the degree of achievement of the task goal was lower.

To see the control feature, simulation results of the loading task when delay time is 0 ms, are presented in Fig. 11. From this figure, we can see that distance errors between current hand position and desired state $(-50, 60)$ are less than 4 cm until the object is loaded on. However, after the contact at $t = 0$ [s], postural control can be said to fail according to the distance errors. In this case, pre-increased motor commands (Fig. 11) were not observed because the controller did not have information to use in advance. The FDM functions as a trigger and prepares dynamics changes that occur. This result suggests that utilizing the FDM contributed to stabilization of the control in the biological system where sensory delays exist.

5.2 Other Computational Models for Loading

Many optimization models that successfully generate stereotypical profiles for reaching movements have been proposed [1]–[3]. Since the models are suggested to explain arm movements, we can think of using the same optimization framework for another arm task. We chose a minimum-variance model to apply it to a loading task [3]. In the minimum-variance model, a trajectory is selected to minimize the variance of the arm position with an assumption that the neural control signals are corrupted by noise whose variance increases with the size of the control signal. During the loading task, the dynamics of the arm changes after loading, making the hand move from a desired position and causing a large positional variance. Therefore, the minimum-variance model is expected to perform the loading

task using the same criterion. Since the ability of maintaining a position is required before performing the loading task, a postural control at $(\theta_S, \theta_E) = (-50, 65)$ was performed at first. Overall, we used the same criterion as in [3] except for variance in the post-movement time. Instead, we used variance integrated over the postural control. In reaching movements, the aim of movements is to reach the end point and stay the hand at that point. But, in the loading task, the aim is to maintain the position throughout the postural control. Therefore, the integrated variance is an appropriate cost function. The details of the optimization method are described in Appendix B. Simulation results are shown in Fig. 12. These results indicate that with the criterion of the minimum-variance model, it isn't able to maintain the posture in the sagittal plane. According to the criterion of the minimum-variance model, the best way of minimizing the positional variance is to make motor commands 0 in the movement time. However, it is not possible, with 0 motor commands under the influence of gravity, to make the hand stay at the desired position. As a result, the method of minimizing the positional variance is to occasionally generate motor commands near 0 (Fig. 12 (b)) in order to resist gravity and reach the desired position at the end of the movement time.

There is another computational model applicable to the loading task called the MOSAIC model [25]. The model is a modular architecture that has multiple pairs of forward and inverse models for motor learning and control. Due to the modular architecture, the model is able to perform motor tasks under many different and uncertain environmental conditions by utilizing forward models and responsibility predictors to determine paired modules appropriate to a situation and their contributions in control. In their study, they simulated an arm tracking task in which the dynamics of the environment periodically switched and presented that adaptive motor behaviors can be achieved according to sim-

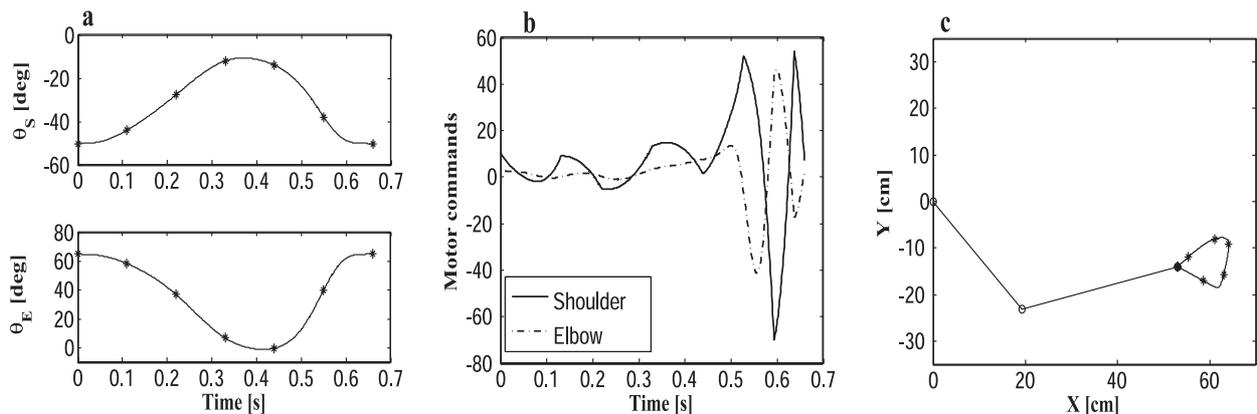


Fig. 12 Simulation results of postural control in the sagittal plane at the desired state $(-50, 65)$ using a minimum-variance model. "*" in (a) and (c) indicates spline knots (see in Appendix B for the optimization method): (a) Optimal trajectories for the shoulder, θ_S , and the elbow, θ_E in the joint coordinate system that minimize positional variance integrated over the postural control. (b) Motor commands in two second-order linear muscles, acting on the shoulder and elbow joint of a two-link arm, with time constants of 92.6 and 60.5 ms. (c) Optimal trajectory in the cartesian coordinate system corresponding to (a).

ulation results of stable tracking control. However, the effect of sensory delays was not considered in their model. Hence, in order to perform a loading task in the MOSAIC model for biological systems that has sensory delays, the degree of contributions of paired modules in motor control should be changed before loading using contextual signals. However, if the timing of the contributions changes are inaccurate, the motor signals from weighted outputs of the inverse models result in distance errors. The distance errors should be modified by the feedback controller for stable control. But, it would be difficult with delayed feedback to successfully compensate the distance error. As a result, the usage of forward models in the sensory feedback path could be considered valid for motor control of the loading task.

Another computational model, using optimal feedback theory, adopts a forward model in the feedback path [26]. It generates motor signals for given tasks based on estimated state variables from the forward model, which has efferent copies of motor signals and delayed sensory feedback as input signals. The theory has been used to explain various motor behaviours (e.g. planar arm movements, ping-pong task) [26]. However, there are several problems that would occur if the optimal feedback theory were applied to the loading task. First, the output of the forward model would need to be changed before loading using contextual signals. Otherwise, it would not be possible to cope with sudden dynamics changes successfully because of sensory delays in biological systems, and pre-increased motor signals before loading would not be observed [4]. Secondly, the optimal control law would need to be recalculated every time the postural position changes since the calculation of optimal control law requires dynamics knowledge and linearization in complex non-linear systems. In this case, high control accuracy is expected, but the required cost is also very high. On the other hand, the control accuracy of the proposed model would likely not be as high because the feedback controller of the proposed model is not optimized to a position as in the optimal feedback theory. In the proposed model, the inverse model allows the hand to maintain near goal positions and lets the feedback controller correct the remaining errors. However, this appears reasonable since the control ability of humans seems to allow and ignore small centimeter order postural distance errors [4]. In addition, our model can acquire control laws without prior knowledge of arm dynamics. However, a merit to using the optimal feedback theory as a motor control model is that the delayed sensory feedback is directly used in generating feedback law. The necessity of online feedback control is stressed and supported by arm moving tasks through targets both in simulations and human subject experiments [26]. Therefore, the proposed model should take into account the delayed feedback not only in acquiring the forward model [14] but also in changing the motor signals [26]. The Smith predictor model is one way to use the sensory feedback directly [6]. The other required improvement in our model is to use an impedance control mechanism. The impedance control method is a simple and useful way to bear sudden

dynamics changes. To implement impedance control in the proposed model, we need to know how humans control the impedance of their muscles. Humans can control impedance voluntarily. We also sometimes show high impedance and co-contractions when surprised or nervous, which we believe can be caused by differences between real results and expected results when doing motor tasks. From these facts, we theorize that there is an independent module that can adjust the impedance level. The module would have at least two inputs. One is humans intention, which can change the impedance level voluntarily. The other is the difference between the sensory feedback information and the predicted information from the forward model.

6. Conclusions

We introduced a learning and control model of the arm for a loading task in the sagittal plane. An Actor-Critic method is adapted in a feedback-error-learning scheme as a feedback controller. FDM is used in a feedback path to predict state and overcome sensory delays. Simply, the loading task can be performed if the stiffness of the muscles is kept high during the task. However, it is not a control policy that humans take. Humans gradually increase the motor commands before the environment changes to stabilize the hand trajectory because of sensory delays [4]. With this model, it was possible to perform the loading task in a human-like way, and tendencies of hand trajectories and hand force similar to the psychophysical experiment were observed. In addition, we showed that stable control in the loading task was possible due to the usage of the FDM.

The proposed framework used the feedback-error-learning scheme proposed as a computational model of cerebellar motor learning [11], [27] and the Actor-Critic method proposed as a model of learning in the basal ganglia [28], [29]. However, we are not sure that the cerebellum and basal ganglia are directly related to performing the loading task, although those regions are possibly used to train the inverse model and the feedback controller, which are considered necessary for the loading task. Psychophysical experiments that measure brain activity (e.g. fMRI) during the loading task would help to identify the brain circuitry and understand the control mechanism.

In our previous study, we showed that the proposed framework can be applied to reaching movements in the sagittal plane and can successfully predict its profiles [14]. Through the simulation of the loading task, it is shown that a unified framework is applicable not only to reaching movements but also to postural control. We believe our findings will help to generalize motor control theories of the arm.

Acknowledgments

Part of this research was supported by Japan Science and Technology Agency CREST.

References

- [1] T. Flash and N. Hogan, "The coordination of arm movements: An experimentally confirmed mathematical model," *J. Neurosci.*, vol.5, no.7, pp.1688–1703, July 1985.
- [2] Y. Uno, M. Kawato, and R. Suzuki, "Formation and control of optimal trajectory in human multijoint arm movement," *Biol. Cybern.*, vol.61, no.2, pp.89–101, 1989.
- [3] C.M. Harris and D.M. Wolpert, "Signal-dependent noise determines motor planning," *Nature*, vol.394, no.6695, pp.780–784, Aug. 1998.
- [4] Y. Tateishi, D. Shin, J. Kim, M. Sato, and Y. Koike, "The measurement of load-on-task performance for the motor control," IEICE Technical Report, NC2004-134, Jan. 2005.
- [5] M. Kawato, "Internal models for motor control and trajectory planning," *Curr. Opin. Neurobiol.*, vol.9, no.6, pp.718–727, Dec. 1999.
- [6] R.C. Miall and D.M. Wolpert, "Forward models for physiological motor control," *Neural Netw.*, vol.9, no.8, pp.1265–1279, Nov. 1996.
- [7] R.S. Johansson and K.J. Cole, "Sensory-motor coordination during grasping and manipulative actions," *Curr. Opin. Neurobiol.*, vol.2, no.6, pp.815–823, Dec. 1992.
- [8] J.R. Flanagan and A.M. Wing, "The role of internal models in motion planning and control: Evidence from grip force adjustments during movements of hand-held loads," *J. Neurosci.*, vol.17, no.4, pp.1519–1528, Feb. 1997.
- [9] M.I. Jordan and D.E. Rumelhart, "Forward models: Supervised learning with a distal teacher," *Cognitive Science*, vol.16, no.3, pp.307–354, July 1992.
- [10] D.M. Wolpert, Z. Ghahramani, and M.I. Jordan, "An internal model for sensorimotor integration," *Science*, vol.269, no.5232, pp.1880–1882, Sept. 1995.
- [11] M. Kawato, K. Furukawa, and R. Suzuki, "A hierarchical neural-network model for control and learning of voluntary movement," *Biol. Cybern.*, vol.57, no.3, pp.169–185, 1987.
- [12] J.S. Albus, "A new approach to manipulator control: The cerebellar model articulation controller (CMAC)," *J. Dyn. Syst. Meas. Control*, Transactions of ASME, vol.97, no.3, pp.220–227, Sept. 1975.
- [13] R.S. Sutton and A.G. Barto, *Reinforcement Learning: An Introduction*, MIT Press, ISBN 0-262-19398-1, 1998.
- [14] H. Kambara, K.S. Kim, D. Shin, M. Sato, and Y. Koike, "Motor control-learning model for reaching movements," *International Joint Conf. on Neural Networks*, pp.555–562, Vancouver, Canada, July 2006.
- [15] J. Izawa, T. Kondo, and K. Ito, "Biological arm motion through reinforcement learning," *Biol. Cybern.*, vol.91, no.1, pp.10–22, July 2004.
- [16] K. Doya, "Reinforcement learning in continuous time and space," *Neural Comput.*, vol.12, no.1, pp.219–245, Jan. 2000.
- [17] Y. Koike and M. Kawato, "Estimation of dynamic joint torques and trajectory formation from surface electromyography signals using a neural network model," *Biol. Cybern.*, vol.73, no.4, pp.291–300, Sept. 1995.
- [18] J.R. Flanagan, P. Vetter, R.S. Johansson, and D.M. Wolpert, "Prediction precedes control in motor learning," *Curr. Biol.*, vol.13, no.2, pp.146–150, Jan. 2003.
- [19] A. Mannard and R.B. Stein, "Determination of the frequency response of isometric soleus muscle in the cat using random nerve stimulation," *J. Physiol.*, vol.229, no.2, pp.275–296, March 1973.
- [20] M. Katayama and M. Kawato, "Virtual trajectory and stiffness ellipse during multijoint arm movement predicted by neural inverse models," *Biol. Cybern.*, vol.69, no.5-6, pp.353–362, 1993.
- [21] H. Kambara, J. Kim, M. Sato, and Y. Koike, "Learning arm's posture control using reinforcement learning and feedback-error-learning," *IEICE Trans. Inf. & Syst. (Japanese Edition)*, vol.J89-D, no.5, pp.1036–1048, May 2006.
- [22] K.S. Kim, H. Kambara, D. Shin, M. Sato, and Y. Koike, "Learning and control model of arm posture," *Proc. 9th Int. Conf. on Intelligent Autonomous Systems*, pp.938–945, Tokyo, Japan, March 2006.
- [23] S.P. Harshfield and D.C. DeHardt, "Weight judgement as a function of apparent density of objects," *Psychonomic Science*, vol.20, pp.365–366, 1970.
- [24] A. Charpentier, "Analyse experimentale de quelques elements de la sensation de poids," *Arch. Physiol. Norm. Pathol.*, vol.3, pp.122–135, 1891.
- [25] M. Haruno, D.M. Wolpert, and M. Kawato, "Mosaic model for sensorimotor learning and control," *Neural Comput.*, vol.13, no.10, pp.2201–2220, Oct. 2001.
- [26] E. Todorov and M.I. Jordan, "Optimal feedback control as a theory of motor coordination," *Nature Neuroscience*, vol.5, no.11, pp.1226–1235, Nov. 2002.
- [27] D.M. Wolpert, R.C. Miall, and M. Kawato, "Internal models in the cerebellum," *Trends in Cognitive Sciences*, vol.2, no.9, pp.338–347, Sept. 1998.
- [28] A.G. Barto, "Adaptive critics and the basal ganglia," in *Models of Information Processing in the Basal Ganglia*, eds. J.C. Houk, J.L. Davis, and D.G. Beiser, pp.215–232, MIT Press, ISBN 0-262-08234-9, 1995.
- [29] K. Doya, "What are the computations of the cerebellum, the basal ganglia and the cerebral cortex?," *Neural Netw.*, vol.12, no.7-8, pp.961–974, Oct. 1999.
- [30] H. Miyamoto, E. Nakano, D.M. Wolpert, and M. Kawato, "TOPS (task optimization in the presence of signal-dependent noise) model," *Syst. Comput. Japan*, vol.35, no.11, pp.48–58, Oct. 2004.

Appendix A: Calculation of Dynamics and Parameters in the Musculoskeletal System

In the simulation, the dynamics equations of the two-link arm in Fig. A·1 are derived using Lagrange's equation of motion [21] and solved with the Runge-Kutta method. The equations are as follows

$$\begin{aligned}
 \tau_S &= \left[I_1 + I_2 + m_2(l_1^2 + 2l_1l_{g2} \cos(\theta_E)) \right] \ddot{\theta}_S \\
 &\quad + \left[I_2 + m_2l_1l_{g2} \cos(\theta_E) \right] \ddot{\theta}_E \\
 &\quad - m_2l_1l_{g2} \sin(\theta_E)(2\dot{\theta}_S\dot{\theta}_E + \dot{\theta}_E^2) \\
 &\quad + (m_1\hat{g}l_{g1} + m_2\hat{g}l_1) \cos(\theta_S) + m_2\hat{g}l_{g2} \cos(\theta_S + \theta_E) \\
 \tau_E &= \left[I_2 + m_2l_1l_{g2} \cos(\theta_E) \right] \ddot{\theta}_S + I_2\ddot{\theta}_E \\
 &\quad + m_2l_1l_{g2} \sin(\theta_E)\dot{\theta}_S^2 \\
 &\quad + m_2\hat{g}l_{g2} \cos(\theta_S + \theta_E)
 \end{aligned} \tag{A·1}$$

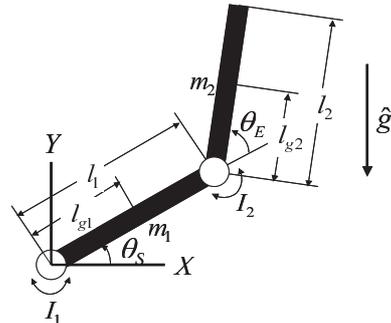


Fig. A·1 2-link arm.

Here, τ_j , θ_j , $\dot{\theta}_j$ and $\ddot{\theta}_j$ indicate joint torque, angle, velocity and acceleration of the arm, respectively. Subscripts ($j = S, E$) indicate the shoulder and elbow. m_i , l_i , l_{gi} and I_i are the weight of the arm, the length of each link, the length from the joint to the mass center of each link, and the moment of inertia around each joint. These parameters are defined as shown in Table A·1.

Besides, muscle parameters in the musculoskeletal system are defined as shown in Table A·2 and Table A·3. k_1 and b_1 are the elastic and viscosity coefficients, respectively. k_0 and b_0 are the intrinsic elasticity and viscosity. l_0 is the intrinsic rest length. l_1 is the muscle length coefficient.

When an object is loaded on the hand, the joint torques for the object (τ_s^{obj} and τ_e^{obj} in Eq. (16)) are added to the right side of Eq. (A·1). Thus torque, the left side of Eq. (A·1), is required more when holding the object than when the object is not loaded to maintain the same posture. The FDM is simulated using the dynamics equations. The process is as follows. Firstly, the FDM judges from the context time, $t^{contact} - m\Delta t$ whether the object should be considered in the control mechanism (i.e. when the time t passed over the context time, the object torque of Eq. (16) is considered. Otherwise, the object torque is set to 0). Secondly, torque in the arm at time t are calculated from the motor commands, \tilde{u}_i^{sum} using Eq. (14) and Eq. (15). Then, the state of the arm is updated with the Runge-Kutta method and the time changes from t to $t + \Delta t$. This calculation is repeated until the time becomes $t + n\Delta t$. During the repeated calculations, the motor command \tilde{u}_i^{sum} is fixed on the value at time t .

Table A·1 Parameters of the 2-link arm.

	Link 1	Link 2
m_i [kg]	1.59	1.44
l_i [m]	0.3	0.35
l_{gi} [m]	0.18	0.21
I_i [kg m ²]	6.78×10^{-2}	7.99×10^{-2}

Table A·2 Moment arms.

	a [m]	
	Around Shoulder	Around Elbow
$sh\ flx$	0.04	0
$sh\ ext$	-0.04	0
$el\ flx$	0	0.025
$el\ ext$	0	-0.025
$dj\ flx$	0.028	0.028
$dj\ ext$	-0.035	-0.035

Table A·3 Muscle parameters.

	k_0 [N/m]	k_1 [N/m]	b_0 [Ns/m]	b_1 [Ns/m]	l_0 [m]	l_1 [m]
$sh\ flx$	1000	6000	50	100	0.150	0.150
$sh\ ext$	1000	4000	50	100	0.055	0.150
$el\ flx$	600	2800	50	100	0.100	0.150
$el\ ext$	600	2400	50	100	0.040	0.150
$dj\ flx$	300	1200	50	100	0.250	0.150
$dj\ ext$	300	1200	50	100	0.130	0.150

Appendix B: Minimum-Variance Model for Postural Control in the Sagittal Plane

In the simulation of the minimum-variance model, the optimal trajectories were found numerically. For the musculoskeletal model, we used the two-link arm in Eq. (A·1) and two linear second-order muscles acting on the shoulder and elbow joint with time constants of $T_1 = 92.6$ ms and $T_2 = 60.5$ ms in Eq. (A·2). The time constants are derived from Eq. (9). The motor command, u_j , of each joint is

$$u_j = T_1 T_2 \ddot{\tau}_j + (T_1 + T_2) \dot{\tau}_j + \tau_j \quad [j = S, E.] \quad (\text{A} \cdot 2)$$

We changed the values of the parameters in the arm and muscles of [3] to make them close to the parameters used in our framework. For the optimization calculation, we used unscented filter and 5th-order splines due to their advantages [30]. First, compared to the Monte carlo method, the unscented filter is simple in its procedure, highly accurate in the estimations of mean and variance, and extremely low cost in computation. Second, the 5th-order splines are used to solve the errors between trajectory generated from the splines and trajectory from motor commands. When 3rd-order splines are used, the accuracy of the motor commands calculated from the inverse dynamics functions of Eq. (A·1) and Eq. (A·2) are decreased because of the low order.

For postural control in the sagittal plane of (x,y), 7 cartesian knots which are evenly spaced in time are used and their first and last knots are fixed at corresponding positions of the desired state $(\theta_S, \theta_E) = (-50, 65)$ with velocity and acceleration of zero. The optimization was simulated to minimize the variance of C_v , integrated over the postural control, with signal-dependent noise. The duration of the movement and control rate were set to 660 ms and 1 ms, respectively. After the cartesian knots are initialized at the desired positions, the knots are adjusted using the simplex algorithm. The procedure of optimization is as follows: (1) Using the inverse kinematics of the arm, the cartesian knots are converted to the knots in the joint coordinate system. (2) The joint knots are parameterized as 5th-order splines. (3) From the splines, motor commands are calculated using the inverse dynamics functions of Eq. (A·1) and Eq. (A·2). (4) Based on the motor commands with signal-dependent noise, the mean and variance of trajectories during the movement are calculated using the unscented filter. (5) The cartesian knots are updated in the direction of minimizing the variance of C_v .



Kyoungsik Kim received his B.S. degree in electronic engineering and avionics from the Korea Aerospace University in 2003. He received his M.S. degree in computational intelligence and systems science from the Tokyo Institute of Technology in 2005. He is currently a Ph.D. student at the Tokyo Institute of Technology. His current research interests include neuroscience and machine learning.



Hiroyuki Kambara received his B.S. degree in computer science and M.S. and Ph.D. degrees in engineering from the Tokyo Institute of Technology in 2002, 2004, and 2007, respectively. He is now an assistant prof. of the Precision and Intelligence laboratory at the Tokyo Institute of Technology. His research interests include neural motor control and motor learning.



Duk Shin received his Ph.D. degree in engineering from the Tokyo Institute of Technology in 2005. He was a researcher at Core Research for Evolution Science and Technology (CREST), JST in 2005–2006. He was an assistant prof. of the Precision and Intelligence Laboratory at the Tokyo Institute of Technology in 2006–2007. He is currently a visiting researcher at TOYOTA CRDL., Inc. His research interests include computational brain science.



Yasuharu Koike received his B.S., M.S., and Dr. Eng. degrees from the Tokyo Institute of Technology in 1987, 1989 and 1996, respectively. From 1989 to 1998, he worked at the Toyota Motor Corporation. From 1991 to 1994, he transferred to Advanced Telecommunications Research (ATR) Human Information Processing Laboratories. In 1998, he moved to the Precision and Intelligence Laboratory at the Tokyo Institute of Technology where he is currently an associate professor. He was a

researcher of Precusory Research for Embryonic Science and Technology, JST from 2000 to 2004 and has been a researcher of CREST, JST since 2004. His research interests include human motor control theory, human interfaces, reinforcement learning and their applications. He is a member of the Society for Neuroscience, VRSJ and JNNS.