Proceedings of the 2003 IEEE
International Conference on Robotics & Automation
Taipei, Taiwan, September 14-19, 2003

Motor Learning Model using Reinforcement Learning with Neural Internal Model

Jun Izawa, Toshiyuki Kondo, and Koji Ito
Department of Computational Intelligence and Systems
Science, Tokyo Institute of Technology(TIT)
4259, Nagatuta Midori, Yokohama Kanagawa 226-8502,Japan.
e-mail)izawa@ito.dis.titech.ac.jp

Abstract— **The present paper proposes a learning control method for the musculoskeletal system of arm based on reinforcement learning. An optimization for the hand trajectory and muscle's force distribution is needed to acquire the reaching motion. The proposed architecture can acquire an optimized motion through learning the task. However, the biological control system composed of musculoskeletal system is not able to sense the state without time delay. The time delay causes instability of learning. The proposed scheme consists of the reinforcement learning part and neural internal model. Neural internal model is employed to compensate for the time delay by estimating the state of muscloskeletay system. Then, there must be a modeling error if some noise is included. Thus we introduce the minimum modeling error criterion for reinforcement learning, which gives not only the reduction of total muscle level but also the smoothness of the hand trajectory. The effectiveness and the biological plausibility of the present model is demonstrated by several simulations.**

I. INTRODUCTION

Recently, reinforcement learning attracts attention as a learning method including a planning of movements[5][6][1]. J.C.Houk et al.[3] have proposed the hypotheses that the basal ganglia might involves actor-critic learning which is a kind of reinforcement learning. Moreover, some researchers on human movement analysis believe that reinforcement learning is crucial for understanding human movements[4]. It is difficult, however, to apply reinforcement learning to the system with any hidden state of the environment, that is, with non-Markov property. Biological control systems include the long time delay (at least 100ms) associated with neural transmission and neural computation, which means non-Markov property.

Now we propose on interacting motor learning method based on reinforcement learning, which introduces a neural internal model to compensate for the time delay. The internal model predicts the state of the environment, with learning the environmental dynamics.

II. MUSCULOSKELETAL SYSTEM

Now, we assume that the muscle force $T \in R^n$ is modeled as

$$T(l,\dot{l},u) = K(u)(l - l_r(u)) + B(u)\dot{l}, \tag{1}$$

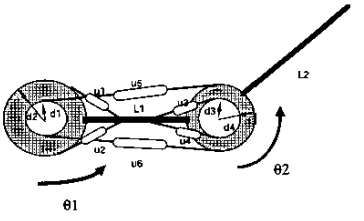


Fig. 2. Musculoskeletal arm

$$K(u) = diag(k_0 + k_i u_i), \tag{2}$$
$$B(u) = diag(b_0 + b_i u_i), \tag{3}$$
$$l_r(u) = [l_{r1}, l_{r2}, l_{r3}, \cdots, l_{rm}]^T, \tag{4}$$
$$l_{ri} = l_0 + r u_i, \tag{5}$$

where K,B are the coefficient matrices of elasticity and viscosity, respectively u is the muscle activation vector, m is the dimension of the muscle motor command vector, l is the muscle length vector, and l_r is the equilibrium length vector of the muscle. The relation of the infinitesimal displacements between the muscle length $l = (l_1, l_2, \cdots, l_m)^T$ and the joint angle $\theta = (\theta_1, \theta_2, \cdots, \theta_j)^T$ is given by

$$dl = G(\theta)d\theta, \tag{6}$$

where G is the Jacobian matrix.

Then, from the principle of virtual work, the relation of the joint torque τ and the muscle tension T is obtained as follows.

$$\tau = -G^T T. \tag{7}$$

We adopt , as an example for application, the musculoskeletal arm which consists of a two link arm with two joints and six muscles, where mono and bi-articular muscles are embedded as shown in Fig.2. The model is simple, but has the essential feature of a musculoskeletal system. Assuming that the moment arm on the adhesion point of the muscle is constant, we can obtain Jacobian G as follows.

$$G = \begin{bmatrix} -d_1 & d_2 & 0 & 0 & -d_5 & d_6 \\ 0 & 0 & -d_3 & d_4 & -d_7 & d_8 \end{bmatrix}^T. \tag{8}$$

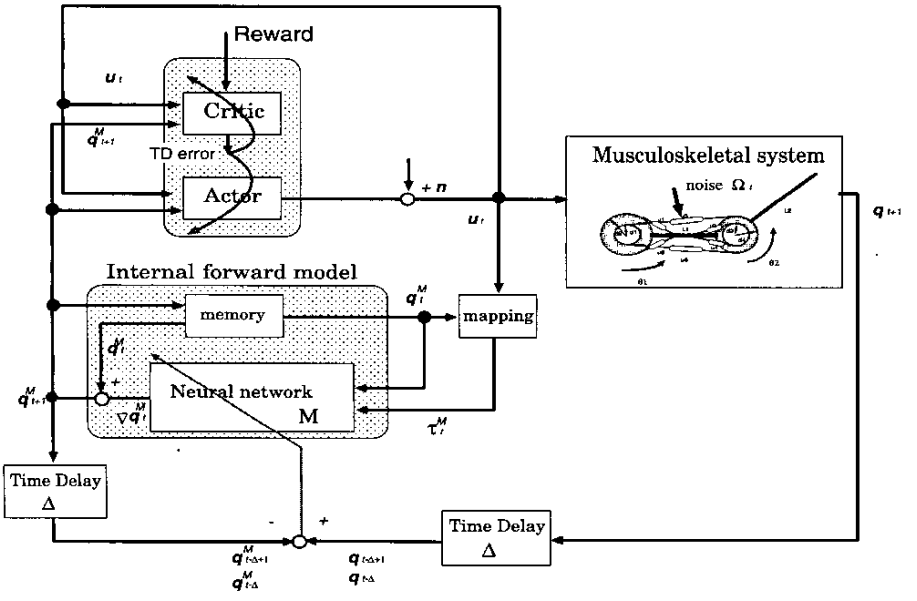


Fig. 1. Block diagram of proposed learning system. The diagram consists of the actor-critic network, musculoskeletal system of arm and internal models are embedded.

From (6) and (7), we obtain

$$\tau = G^T K_m G[-r(G^-)u - \theta] - G^T B_m G\dot{\theta} \quad (9)$$
$$= R(u)[\theta_v(u) - \theta] - D(u)\dot{\theta}, \quad (10)$$

where R and D are the coefficient matrices of elasticity and viscosity in the joint spaces, and θ_v is called the virtual equilibrium point.

A moment arm is defined as follows. Muscle 1 and 2 is 4.0cm. Muscle 3 and 4 is 2.5cm. Muscle 5 and 6 is 3.5cm. A parameter of muscle model is defined so that k, k_0, b, b_0, r are 100N/m, 2021N/m,100Ns/m,200Ns/m and 0.2 respectively.

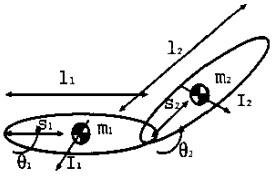


Fig. 3. Two joint arm

Dynamic equation of 2 link arm (Fig.3) is

$$\begin{aligned}\tau_1 &= M_{11}\ddot{\theta}_1 + M_{12}\ddot{\theta}_2 + h_{122}\dot{\theta}_2 + 2h_{112}\dot{\theta}_1\dot{\theta}_2 \\ \tau_2 &= M_{21}\ddot{\theta}_1 + M22\ddot{\theta}_2 + h_{211}\dot{\theta}_1, \end{aligned} \quad (11)$$

where,

$$\begin{aligned}M_{11} &= m_1 s_1^2 + I_1 + m_2(l_1^2 + s_2^2 + 2l_1 s_2 \cos\theta_2) + I_2 \\ M_{12} &= M_{21} = m_2(s_2^2 + l_1 s_2 \cos\theta_2) + I_2 \\ M_{22} &= m_2 s_2^2 + I_2 \\ h_{122} &= h_{112} = -h_{211} = -m_2 l_1 s_2 \sin\theta_2. \end{aligned} \quad (12)$$

Each parameter is shown in Tab.I

TABLE I
PARAMETERS OF TWO-LINK ARM

	Link1	Link2
m_i[kg]	1.59	1.44
l_i[m]	0.35	0.35
s_i[m]	0.18	0.21
I_i[kg·m^2]	1.63×10^{-2}	1.64×10^{-2}

III. SYSTEM ARCHITECTURE

Fig.1 shows an overview of the proposed architecture. The schematic diagram consists of Actor-Critic network[2], musculoskeletal system, and internal forward model.

The musculoskeletal system receives a motor command $u_t \in R^n$ and outputs the joint angles and angular velocities $q_{t+1} = (\theta_{t+1}, \dot{\theta}_{t+1}) \in R^m$ as the state values of the system. The motor command u_t consists of the output of the Actor-network and the search noise. Another noise Ω_t derived from a fluctuation in motor neurons is added to the motor

command. The internal forward model predicts the state of the system q^M_{t+1} with receiving the motor command u_t. The state in the internal forward model q^M_t is an estimated value of q_t. The output of the neural network ∇q^M_t is a variation of the state value at each step. Thus, the state of the next step q^M_{t+1} is calculated by adding the variation ∇q^M_t to the current state q^M_t. The state of the musculoskeletal system is observed with a time delay Δ. The internal forward model consists of the neural network, the memory for current states and the mapping from the motor command to the joint torque.

The mapping from the motor command u_t to the joint torque τ^M_t is calculated based on Eq.(10). The weight parameter of the Actor-Critic network is updated so as to decrease the TD-error.

The state value received by the Actor-Critic is defined as s and distinct from the state value of the musculoskeletal system q because, when $n > m$, the system dose not obtain an optimal value through reinforcement learning based on only q. Thus, in the present paper, reinforcement learning is executed receiving $s_t = (q^M_t, u_{t-1})$ as a state value.

IV. Reinforcement Learning with the Internal Forward Model

We applied the reinforcement learning with the internal forward model to the musculoskeletal system. Fig.2 shows an overview of the algorithm. The musculoskeletal system consists of two-link rigid body and six muscle-like actuators. The muscle model has the spring-like property and generates a muscle force by moving a rest length of the spring. It is, here, assumed that the insertion point of the muscle is constant irrespective of the change of the joint angle.

The initial position of the hand is adjusted such that the end point is at the start position defined in Fig.5. The initial value of u is preset to 0. The internal forward model includes the neural network(NN) whose units consists of sigmoidal function. The number of unit in middle layer of the neural network is twelve. The updating ratio of the network is 0.1. The Actor-Critic network is made of Gauss-Sigmoid NN[7]. The number of units in input layer of sigmoidal neural network embedded in Gauss-Sigmoid NN is 86. The number of units in meddle layer is twelve in Critic-network and is eighteen in Actor-network.

The initial weight of neural network is preset randomly except the weight between the middle layer and output layer which is preset to 0 except for bias unit. The weight related to bias unit is preset to 1.

The discount rate of the value function in reinforcement learning is 0.99. The search noise within the motor command n is made of low-pass filtered white noise with the variance σ .The low-pass filter is $\tau\dot{n} = -n + n_{in}$, where the time constant $\tau = 0.1$. The deviation of the noise is

Initialize the critic network and the actor network
Repeat(for each trial):

1) Initialize $s^M_0 = (q^M_0, u_0)$
2) Repeat(for each step)
   a) Observe $s^M_t = (q^M_t, u_{t-1})$
   b) $u_t \leftarrow A(s^M_t, w_a) + n_t$
   c) Calculate the state of the system and the forward model,

$$q_{t+1} \leftarrow P(q_t, u_t)$$
$$q^M_{t+1} \leftarrow q^M_t + M(q^M_t, \tau^M_t, w_M)$$

   d) Update the Forward-model-network, M
      input : $q^M_{t-\Delta}, \tau^M_{t-\Delta}$
      target : $q_{t-\Delta+1} - q_{t-\Delta}$
   e) If $t \geq \Delta$, initialize $s^{M\prime}_0 = (q^{M\prime}_0, u'_0)$,start the internal learning process.
      i) Observe $s^{M\prime}_{t-\Delta} = (q^{M\prime}_{t-\Delta}, u'_{t-1-\Delta})$
      ii) $u'_{t-\Delta} \leftarrow A(s^{M\prime}_{t-\Delta}, w_a) + n_{t-\Delta}$
      iii) Calculate the state of the forward model,

$$q^{M\prime}_{t-\Delta+1} \leftarrow q^{M\prime}_{t-\Delta} + M(q^{M\prime}_{t-\Delta}, \tau'_{t-\Delta}, w_M)$$

      iv) Calculate the TD-error,

$$\delta'_{t-\Delta} \leftarrow r'_{t-\Delta+1} + \gamma\hat{V}(s^{M\prime}_{t-\Delta+1}, w_c) - \hat{V}(s^{M\prime}_{t-\Delta}, w_c)$$

      v) Train the Critic-network
         input : $s^{M\prime}_{t-\Delta}$
         target : $\hat{V}(s^{M\prime}_{t-\Delta}, w_c) + \alpha\delta'_{t-\Delta}$
      vi) Train the Actor-network
         input : $s^{M\prime}_{t-\Delta}$
         target : $A(s^{M\prime}_{t-\Delta}, w_a) + \beta\delta'_{t-\Delta} \cdot n_{t-\Delta}$

Fig. 4. Learning algorithm

20% of the output and adjusts to be 1% as the increase of $\hat{V}$.

The reaching motion is selected as an example showing the effectiveness of the architecture. The agent has to move the hand position from the start point to the goal area.

Now, it is assumed that a biological noise is included in the muscle activation. And it is assumed that the variance of the noise is proportional to the squared value of the motor command.

These causes a modeling error, which gives a bad effect on learning. From this, we introduce *minimum modeling error criterion* for the reaching motion. Then, the cost can be expressed by the reward definition as follows.

$$r_E = (x_t - x^M_t)^T (x_t - x^M_t), \tag{13}$$

where, x^M is the estimated state of hand position. As a result, a reward definition in the present paper can be expressed as follows.

$$r = \begin{cases} 1 - c \cdot r_E, & \text{for } x_H \in S \cap x_H \in G \\ 0, & \text{for } x_H \in S \cap x_H \notin G \\ -1, & \text{for } x_H \notin S \end{cases} \quad (14)$$

where, x_H ,S,G and c is the hand position, the working area, the goal area, weighting coefficient respectively.

V. Simulation

A. Learning based on the state prediction

The following results show that the state prediction with the internal forward model is effective for reinforcement learning. Learning of Actor-Critic network was executed in parallel with updating the weight parameter of internal forward model. The reward was given when the end point position reached in goal area (radius is 2cm). In this simulation, the variation in muscle activity and the minimum modeling error criterion is not included. Note that, $c = 0$.

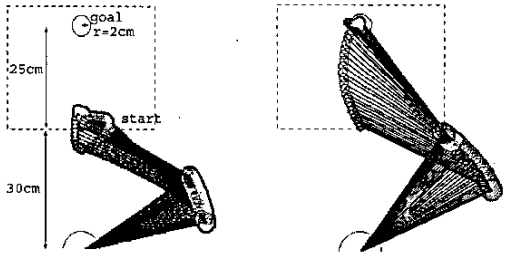
First, the result for the system without time delay is shown in Fig.5. Although the movement of arm seems exploratory in the early stage of learning, the smooth hand path is acquired in the final stage. The learning is exactly converged.

Next, Fig.6 shows the hand paths of the system with time delay. Note that the system has no internal forward model and did not involve the state prediction. In each condition, (a)(b)(c), the system has time time delay in the state observation which is 50ms, 100ms or 200ms respectively. The longer time delay is, the more remarkable the vibration of the end point is. Moreover, learning a reaching motion failed in the case of 200ms.

On the other hand, as shown in Fig.7, the hand path is smooth when the system has the internal forward model to compensate for the time delay. The hand trajectory is smooth without vibration. If the radius of the goal area is smaller, the amplitude of vibration will be reduced, because it seems to be damped into goal area as seen in Fig.7. In addition, Fig.8 shows that the system with the state prediction can get more rewards than the system without the state prediction. Thus, the performance of the learning system can be improved by the state prediction when the system involves time delay in the state observation.

B. Minimum modeling error criterion

Next, we introduced the minimum modeling error criterion into the reinforcement learning with internal forward model. The noise was added to the motor command during leaning. The internal forward model was learned in parallel with reinforcement learning from the initial stage of learning. c in Eqn.14 is preset to 5. Then c is adjusted to be 10 as progress of learning because, if c is preset high in the initial stage, the system become difficult in acquiring enough reward.



(a) 1th trial (b) 30000th trial

Fig. 5. Stick picture of acquired reaching motions:(a) The picture of the 1st trial. At the initial state of the reaching motion, the distance between the shoulder joint and the hand position is 30cm, and the distance between the hand position and the center of target goal is 25cm. The radius of goal area is 2cm.(b)The picture of the 30000th trial. The reaching motion is acquired with roughly straight hand path.

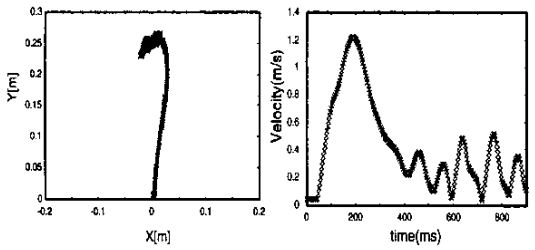
Fig.9 shows the hand path and the tangential velocity after 30000 trials, in which the time delay is 100ms. The hand path is straight and the tangential velocity seems to be bell-shaped. Comparing with Fig.7, the peak of velocity decreases and the velocity profile becomes smooth. That is to say, the minimum modeling error criterion has much effect on the smoothness of hand trajectory.

Fig.10 shows the resultant muscle forces. Although we preset the motor command in the initial stage so that each muscle force can be $250N$ or so, the acquired muscle forces is between $30N$ and $100N$. These indicate that the total muscle force decreased as a result of leaning.

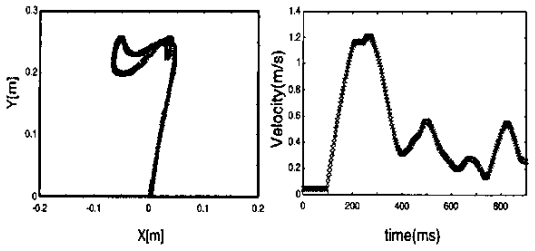
Fig.11 shows the total muscle force of six muscles at each trial. We simulated not only when the variance of the noise in muscle activation was proportional to the squared motor command but also when the variance of the noise was constant. The total muscles force decreases as a trial number when the variance correlates with the motor command. On the other hand, the total muscle force dose not decrease when the variance is constant, which means the system can not acquire the motion keeping the total muscle force low.

Accordingly, the reward given with the minimum modeling error criterion makes the hand path straight and makes tangential velocity smooth with decreasing the total muscle force.

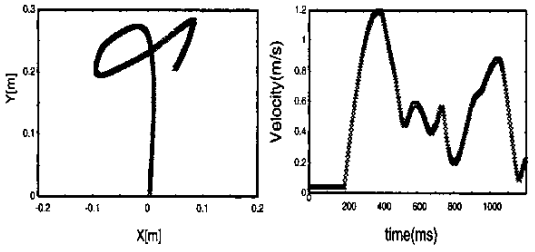
Minimizing the modeling error at the goal area is identical to minimizing the variance of the hand position. Eq.(13) is related to the variance of hand position when the variance of noise added to the muscle activation is proportional to the squared motor command. Minimizing the modeling error is identical to minimizing the weighted squared summation of the motor command through time. This is why the total muscles force decreases. That is to



(a) $\Delta = 50$ms



(b) $\Delta = 100$ms



(c) $\Delta = 200$ms

Fig. 6. Hand trajectory and tangential velocity after 15000 trials without forward model regardless of time delay

say, because the minimum modeling error criterion keeps the total muscle force low, the peak of the hand velocity decreases and the hand path becomes straight.

VI. Conclusion

The proposed method is essentially composed of the internal model and Actor-Critic network. The internal model is made up through the supervised learning in parallel with reinforcement learning. The difficulty caused by the time delay included in the neural transmission is solved by the state prediction with neural internal model. In addition, we proposed the minimum modeling error criterion as a reward definition. This enables the system obtain smooth movements with the low total muscle force which is reasonable for metabolic cost. Until now, some motor learning models with the forward model and

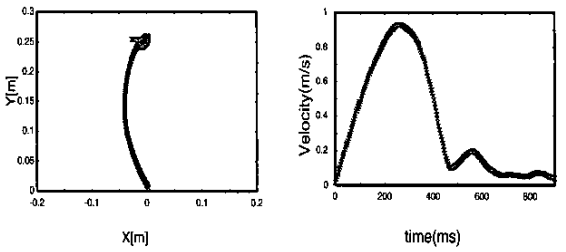
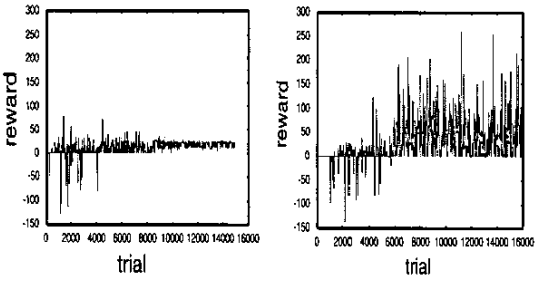


Fig. 7. Hand trajectory and tangential velocity after 30000 trials with the forward model to compensate for the time delay(200ms)



(a) without internal forward model

(b) with internal forward model

Fig. 8. Cumulative reward against trial number. If the reaching motion is realized very well, the cumulative reward per one trial is high level.

reinforcement learning has been proposed. However, those models involve no muscle systems and no time delay in neural transmission from the muscle skeletal systems to the central nervous systems. In the present paper, it is assumed that the time delay is known in advance. In the case of real biological motor learning, it is not probable that the delay time is given. Our future work is to propose a learning model with the unknown time delay.

Acknowledgment

A part of this research was supported by Takahashi Industrial and Economic Research Foundation, Grant-in-Aid(14350277,14750362) for scientific research,JSPS, and Mitutoyo Association for Science and Technology.

VII. References

[1] K. Althoefer, B. Krekelberg, D. Husmeire, and L. Seneviratne. Reinforcement learning in a rule-based navigator for robotic manipulators. *Neurocomputing*, 37:51–70, 2001.

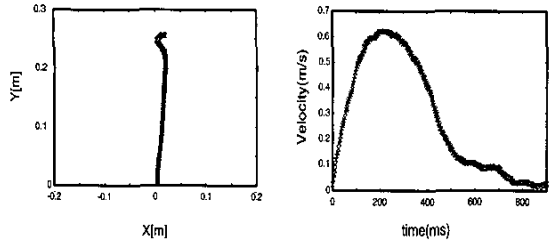


Fig. 9. Hand trajectory and tangential velocity after 30000 trials with the forward model to compensate for the time delay(100ms) and the minimum model prediction error criterion.

[2] A. F. Barto, R. S. Sutton, and C. W. Abderson. Neuronlike adaptive elements that can solve difficult learning control problem. *IEEE Transactions on Systems, Man, and Cybernetics*, 13(5):834–846, 1983.

[3] J. C. Houk, J. L. Adams, and A. G. Barto. A model of how the basal ganglia generate and use neural signals that predict reinforcement. In *Models of Information Processing in the Basal Ganglia*. MIT Press, Canbrigde, MA, USA, 1994.

[4] M.I. Jordan and D.M. Wolpert. Computational motor control. In M.Gazzaniga, editor, *The Cognitive Neurosciences*, pages 601–620. MIT Press, 1999.

[5] P. Martin and J.R. Millan. Learning reaching strategies through reinforcement for a sensor-based manipulator. *Neural Network*, 11:359–376, 1998.

[6] P. Martin and J.R. Millan. Robot arm reaching through neural inversions and reinforcement learning. *Robotics and Autonomous Systems*, 31:227–246, 2000.

[7] Katsunari Shibata, Masanori Sugisaka, and Koji Ito. Hand reaching movement acquired through reinforcement learning. In *Proc. of 2000 KACC (Korea Automatic Control Conf.)*, volume 90rd (CD-ROM), 2000.

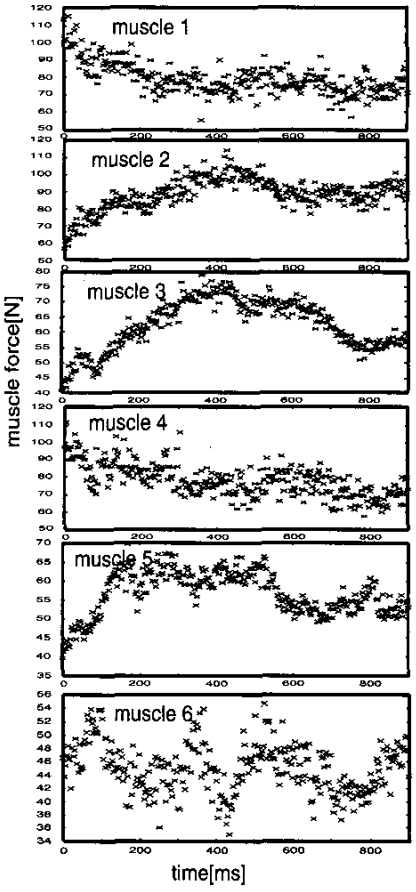


Fig. 10. Muscle force (30000 trials). The number of muscle corresponds to the number in Fig.1. The noise is added to the motor command launched from Actor-network. This gives a variance in muscle force.

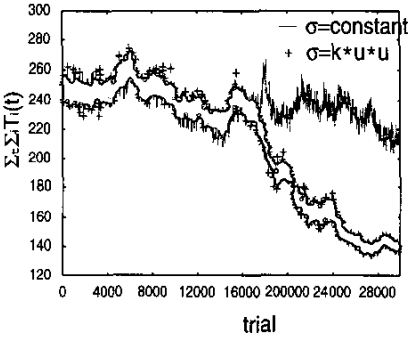


Fig. 11. Total muscle force summation against trials. The line plotted with "-" indicates the total muscle force when the variance of noise is independent from activation level. The line plotted with "+" indicates the total muscle force when the variance of noise depends on activation level.