Proceedings of the 28th IEEE
EMBS Annual International Conference
New York City, USA, Aug 30-Sept 3, 2006

Macroscopic Modeling and Identification of the Human Neuromuscular Network

Yoshihiko Nakamura, Katsu Yamane, and Akihiko Murai

Department of Mechano-Informatics
University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656 Japan
nakamura@ynl.t.u-tokyo.ac.jp

Abstract In this paper, we build a mathematical model of the whole-body neuromuscular network and identify its parameters by optical motion capture, inverse dynamics computation, and statistical analysis. The model includes a skeleton, a musculotendon network, and a neuromuscular network. The skeleton is composed of 155 joints representing the inertial property and mobility of the human body. The musculotendon network includes more than 1000 muscles, tendons, and ligaments modeled as ideal wires with any number of via points. We also develop an inverse dynamics algorithm to estimate the muscle tensions required to perform a given motion sequence. Finally, we model the relationship between the spinal nerve signals and muscle tensions by a neural network. The resulting parameters match well with the agonist-antagonist relationships of muscles. We also demonstrate that we can simulate the patellar tendon reflex using the neuromuscular model. This is the first attempt to build and idenfity a macroscopic model of the human neuromuscular network based only on non-invasive motion measurements, and the result implies that the activation commands from the motor neurons can be considerably simple compared with the number of muscles to be controlled.

Index Terms Musculoskeletal Human Model, Neuromuscular Network Model, Inverse Dynamics, Motion Capture.

I. Introduction

The mechanism for generating and coordinating human motions is still an open research issue. The brain science community has tried for years to understand and model how the brain controls whole-body motions [1], [2]. The biomechanics community, on the other hand, has been working on dynamics computation and motion analysis of musculoskeletal human models [3] [5]. However, there is still a large gap between the two approaches.

This paper extends our previous work on musculskeletal human model [6] by adding a mathematical model of the whole-body neuromuscular network. The parameters of the neuromuscular model are identified through experiments using an optical motion capture system, inverse dynamics computation, and statistical analysis. We expect that this would become the first step towards bridging the gap between two fields. The developed model will also have direct applications in such fields as biomechanics, neurology, rehabilitation, and sport science.

The model includes a skeleton, a musculotendon network, and a neuromuscular network. The neuromuscular system is models as a neural network representing the relationship between the spinal nerve signals and muscle tensions. In order to identify the parameters of the neurons, we first measure a number of different human motions via an optical motion capture system. We then used our musculoskeletal model and the inverse dynamics algorithm to estimate the muscle tensions during the captured motions. Independent component analysis (ICA) of the muscle tensions revealed that the tensions estimated for the measured motions can be represented by about 100 independent signals. We therefore train the neural network model of the neuromuscular system by using independent signals as inputs and corresponding muscle tensions as desired outputs. We analyze the obtained parameters and show that they have distinct correspondence with the agonist/antagonist relationships of the muscles.

The rest of this paper is organized as follows. Section II briefly summarizes the musculoskeletal model presented in [6] and describes some improvements made in this paper. In Section III, we analyze the muscle tension data obtained through experiments and derive our neuromuscular model. We finally present the experimental results in Section IV, followed by concluding remarks.

II. Inverse Dynamics of Musculoskeletal Model

A. The Musculoskeletal Model

The musculoskeletal human model consists of a musculo-tendon network and a skeleton. The skeleton is a set of rigid links connected by mechanical joints, while the musculo-tendon network is composed of the elements to drive and/or constrain the bones including muscles, tendons, ligaments, and cartilages. The model used in this paper is basically the same as the one used in [6] but contains the following improvements:

1-4244-0033-3/06/$20.00 ©2006 IEEE.

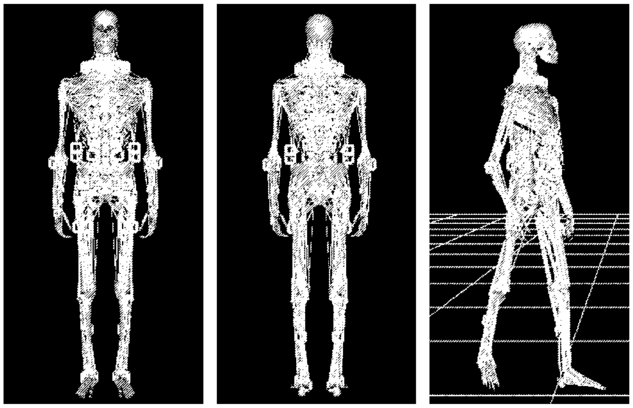

Fig. 1. The musculoskeletal human model used in this paper. Left: front view, center: back view, right: a snapshot from a captured walk motion.

TABLE I

Complexity of the musculoskeletal model.

musculo-tendon network	
muscles	989
tendons	50
ligaments	117
cartilages	34
total wires	1190
muscle groups	78
virtual links	72
skeleton	
bones	200
bone groups	53
total DOF	155

- Many small muscles around the spine are added to reduce the joint torque errors observed in [6]. The number of muscles grew from 366 to 989.
- The muscles are grouped according to their role (for example, extend the left knee), which will be used in the inverse dynamics computation later.

Fig. 1 shows the new model and Table I summarizes the complexity of the model used in this paper.

B. Inverse Dynamics

In [6] we proposed an inverse dynamics algorithm for the musculoskeletal human model, which computes the ground contact forces and muscle tensions by solving the following equation:

$$\boldsymbol{\tau}_G = \boldsymbol{J}^T \boldsymbol{f} + \boldsymbol{J}_C^T \boldsymbol{\tau}_C \tag{1}$$

where

$\boldsymbol{\tau}_G$: generalized forces

$\boldsymbol{J}$: Jacobian matrix of the wire length w.r.t. the generalized coordinates

$\boldsymbol{f}$: wire tensions

$\boldsymbol{J}_C$: Jacobian matrix of the contact points w.r.t. the generalized coordinates

$\boldsymbol{\tau}_C$: ground contact forces.

The solution consists of the following two steps:

1) Compute the ground contact forces $\boldsymbol{\tau}_C$ by only considering the rows of Eq.(1) corresponding to the 6 DOF of the hip joint. We apply quadratic programming to solve the optimization problem.
2) Eliminate $\boldsymbol{\tau}_C$ from Eq.(1):

$$\boldsymbol{\tau}'_G = \boldsymbol{\tau}_G - \boldsymbol{J}_C^T \boldsymbol{\tau}_C = \boldsymbol{J}^T \boldsymbol{f}. \tag{2}$$

and compute the muscle tensions by solving another optimization by either linear programming (LP) or quadratic programming (QP).

In this paper, we adopt the same method as [6] for step 1). The rest of this section describes our improvement of the algorithm for step 2).

In [6] we found that LP solves our complex optimization problem much faster than QP. However, the inherent problem of LP is that the resulting muscle tensions can be both temporally and spatially discontinuous. Spatial discontinuity implies that the tensions of geometrically close muscles can be completely different, which is not likely to happen in human body.

We solve this problem by considering a measure of variation of tensions of the muscles in each group. The new LP formulation is summarized as follows:

For constant vectors with positive components $\boldsymbol{a}_\tau$, $\boldsymbol{a}_f$, and $\boldsymbol{a}_m$, find $\boldsymbol{\delta}_\tau$, $\boldsymbol{\delta}_f$, $\boldsymbol{\delta}_m$, and $\boldsymbol{f}$ that minimize

$$Z = \boldsymbol{a}_\tau^T \boldsymbol{\delta}_\tau + \boldsymbol{a}_f^T \boldsymbol{\delta}_f + \boldsymbol{a}_m^T \boldsymbol{\delta}_m \tag{3}$$

subject to

$$-\boldsymbol{\delta}_\tau \le \boldsymbol{\tau}'_G - \boldsymbol{J}^T \boldsymbol{f} \le \boldsymbol{\delta}_\tau \tag{4}$$

$$\boldsymbol{\delta}_\tau \ge 0 \tag{5}$$

$$-\boldsymbol{\delta}_f \le \boldsymbol{f} - \boldsymbol{f}^* \le \boldsymbol{\delta}_f \tag{6}$$

$$\boldsymbol{\delta}_f \ge 0 \tag{7}$$

$$-\boldsymbol{f}_{max} \le \boldsymbol{f} \le 0 \tag{8}$$

$$-\boldsymbol{\delta}_m \le \boldsymbol{E}_G \boldsymbol{f} \le \boldsymbol{\delta}_m \tag{9}$$

$$\boldsymbol{\delta}_m \ge 0 \tag{10}$$

where the third term of Eq.(3) has been added to consider the muscle tension variation. The detailed description of the equations will be given in the subsequent paragraphs.

The first term of Eq.(3) and Eqs.(4)(5) try to minimize the error of Eq.(2) to assure the physical validity of the result. Instead of including Eq.(2) as an equality condition, we relax the problem because Eq.(2) may not have an exact solution. By including the term $\boldsymbol{a}_T^T \boldsymbol{\delta}_T$ in the objective function Eq.(3), we can obtain the minimum possible values of for elements of $\boldsymbol{\delta}_T$, which are constrained to be positive by the inequality condition Eq.(5). On the other hand, Eq.(4) ensures that the error of Eq.(3) is smaller than $\boldsymbol{\delta}_T$. By combining these constraints, we can minimize the error of Eq.(3).

The second term of Eq.(3) and Eqs.(6)(7) tries to bias $\boldsymbol{f}$ towards a given desired wire tension vector $\boldsymbol{f}^*$. The user can, for example, specify the relationship between a pair of flexor and extensor muscles by supplying appropriate values to $\boldsymbol{f}^*$. For example, this information could be obtained by electromyograph (EMG) measurements for biomechanical applications [7]. Simply setting $\boldsymbol{f}^* = 0$ gives the minimum wire tensions.

Eq.(8) represents the upper and lower bounds of the wire tensions where $\boldsymbol{f}_{max} \geq 0$ is the vector of maximum muscle tensions. The elements of $\boldsymbol{f}_{max}$ can be selected independently for each muscle. We may also consider the muscle length and its velocity to compute $\boldsymbol{f}_{max}$ by Hill's muscle model [8].

Finally, the third term of Eq.(3) and Eqs.(9)(10) are included to equalize as much as possible the tensions of muscles in the same group. In quadratic programming, this term could be replaced by adding squared sum of wire tensions to the evaluation function. Suppose group m includes n_m muscles and $\mathcal{G}_m$ denote the set of their indices. The average tension of group m is computed by

$$\bar{f}_m = \frac{1}{n_m} \sum_{k \in \mathcal{G}_m} f_k \tag{11}$$

where f_k is the tension of the k-th muscle. The difference between the average tension and the k-th ($k \in \mathcal{G}_m$) muscle's tension is

$$\Delta f_{mk} = \bar{f}_m - f_k = \boldsymbol{E}_{Gmk} \boldsymbol{f} \tag{12}$$

where $\boldsymbol{E}_{Gmk}$ is a row vector whose i-th element is $(1 - n_m)/n_m$ if $i = k$, $1/n_m$ if $i \in \mathcal{G}_m$ and $i \neq k$, and 0 otherwise. By collecting $\boldsymbol{E}_{Gmk} (k \in \mathcal{G}_m)$ for all groups and stacking them vertically, we obtain the matrix $\boldsymbol{E}_G$ in Eq.(9).

C. Forward Dynamics

Forward dynamics is the process to compute the joint acceleration from the given wire tensions. Derivation of the algorithm is straightforward because we can easily transform the wire tensions into equivalent joint torques by Eq.(1), and then compute the joint accelerations by applying any forward dynamics algorithm for rigid kinematic chains (e.g. [9], [10]).

D. Experiments with Motion Capture Data

We verified the effect of the improvements by performing inverse dynamics computation for a motion in which the subject bends the spine back and forth. Figures 2 and 3 compare the results of inverse dynamics computations using the model in [6] and the improved one. In each graph, the dashed black line represents the required joint torque obtained by Newton-Euler inverse dynamics computation, while the blue and red lines are the joint torques computed from the muscle tensions obtained by the old and new models, respectively. The new model yields much more precise and smooth muscle tensions.

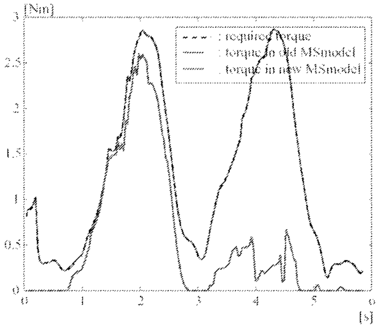

Fig. 2. The joint torque of the third neck vertebra. Black dashed: result of Newton-Euler inverse dynamics computation; blue: computed from the muscle tensions obtained by the old model; red: computed from the muscle tensions obtained by the new model.

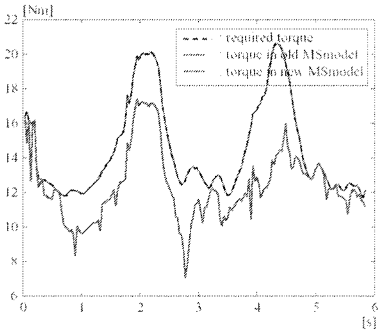

Fig. 3. The joint torque of the sixth rib vertebra. Black dashed: result of Newton-Euler inverse dynamics computation; blue: computed from the muscle tensions obtained by the old model; red: computed from the muscle tensions obtained by the new model.

III. Modeling the Neuromuscular Network

A. Independent Component Analysis of Muscle Tensions

Independent Component Analysis (ICA) [11] is a statistical technique for analyzing observed data. The technique computes hidden factors underlying the observed data, assuming that the data are linear mixtures of nongaussian and mutually independent signals. In contrast to Principal Component Analysis (PCA) which tries to find the common features of a signal, ICA tries to identify the different signal sources.

Our problem here is to find a constant matrix $\boldsymbol{W}_{ICA} \in \boldsymbol{R}^{N_w \times N_{ICA}}$ that maps a vector $\boldsymbol{s} \in \boldsymbol{R}^{N_{ICA}}$ to the estimated wire tensions $\boldsymbol{f} \in \boldsymbol{R}^{N_w}$:

$$\boldsymbol{f} = \boldsymbol{W}_{ICA}\boldsymbol{s} \tag{13}$$

where N_w is the total number of wires ($N_w = 1190$ in our model) and $N_{ICA} (< N_w)$ is the manually specified number of independent signal sources. $\boldsymbol{s}$ represents a set of independent signals required to described the wire tensions of a frame.

We applied ICA to the muscle tension data of 1190 wires estimated for 1344 frames (44.6 seconds) of motion capture data. Fig. 4 show two examples of comparisons of the original and reconstructed wire tensions with $N_{ICA} = 120$. 177 frames from the two examples comparing the original muslce tension and the tension recovered by ICA with 120 independent sources. Fig. 5 shows the average and variance of the error between original and recovered tensions of all wires. These results indicate that the tensions of 1190 wires can be represented by as few as 120 independent signals.

This figure reminds us of the fact that the muscles are governed by 124 left/right and anterior/posterior rami. In this paper, we further divide them into 94 groups by combining those governing the same set of muscles. The ICA in the rest of the paper is performed with $N_{ICA} = 94$.

In order to show the possibility of using $\boldsymbol{s}$ as spinal nerve signal, we first define matrix $\boldsymbol{C} \in \boldsymbol{R}^{N_{ICA} \times N_w}$ whose (i, j)-th element is 1 if i-th ramus is connected to the j-th wire and 0 otherwise. We then compute a square matrix $\boldsymbol{P} \in \boldsymbol{R}^{N_{ICA} \times N_{ICA}}$ by $\boldsymbol{P} = \boldsymbol{C}\boldsymbol{W}_{ICA}$. The (i, j)-th element of $\boldsymbol{P}$ represents the sum of elements of $\boldsymbol{W}$ corresponding to the j-th ICA component and muscles connected to the i-th ramus. Therefore, if there is an one-to-one mapping from ICA to nerve signal, each row and column of $\boldsymbol{P}$ will have only one outstanding peak. Fig. 6 shows the values of several representative rows of $\boldsymbol{P}$, where the horizontal axis indicates the indices of the columns. As expected, most of the lines (rows of $\boldsymbol{P}$) have only one peak at different columns. Fig. 7 marks the column index of the maximum element of each row, where the horizontal and vertical axes represent the indices of the nerves and ICA components, respectively. We can observe that each nerve corresponds to different ICA components, except for those marked by ovals in the figure which corresponds to plexus.

B. Neural Network Model

The ICA discussed in the previous subsection revealed the following important points:

- Tensions of more than 1000 wires required to perform a long sequence of motions can be represented by as few as 100 independent signals.

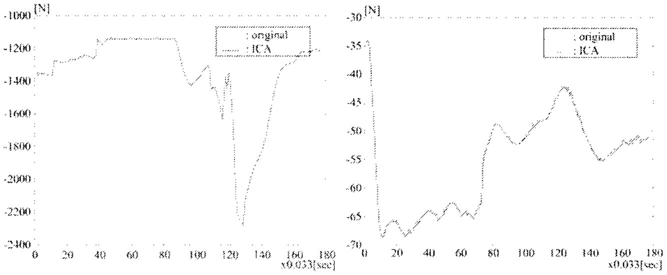

Fig. 4. Original and recovered muscle tensions. Black line: original muscle tensions; red line: muscle tensions recovered by ICA.

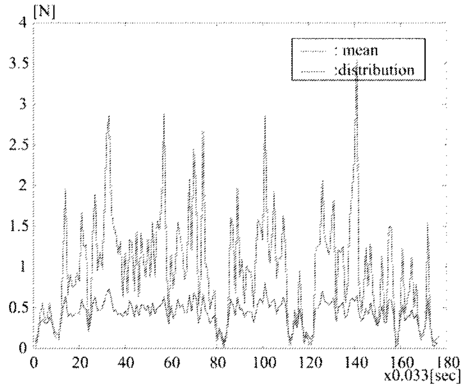

Fig. 5. Average and variance of error between original and recovered tensions of all wires.

- There seems to be an one-to-one correspondence between most of the ICA components and spinal nerves, except for those forming plexus.

Taking these facts into consideration, in this section we build a neural network model which takes spinal nerve signals as inputs and outputs the muscle tensions, and then identify its parameters based on the experimental results.

The actual system contains complex closed loops due to reflex as shown in the top figure of Fig. 8, whose corresponding model would look like the bottom figure. However, it is usually very difficult to identify the system embedded in a closed loop. We therefore convert the closed-loop system to the open-loop system shown in the top figure of Fig. 9 and identify the parameters of its corresponding model (bottom figure of Fig. 9).

Fig. 10 shows the three-layered neural network model. The input layer with N_{ICA} neurons takes the independent signals identified by the ICA as inputs. The middle layer also has N_{ICA} neurons each corresponding to the one of the spinal rami. All neurons in the input and middle layers are interconnected to represent the correspondence between the independent signals and spinal rami. The ouput layer has N_w neurons which output the tensions of

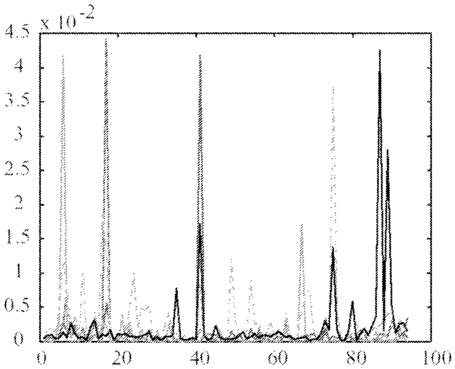

Fig. 6. Values of selected rows of $\boldsymbol{P}$ displayed in the order of column.

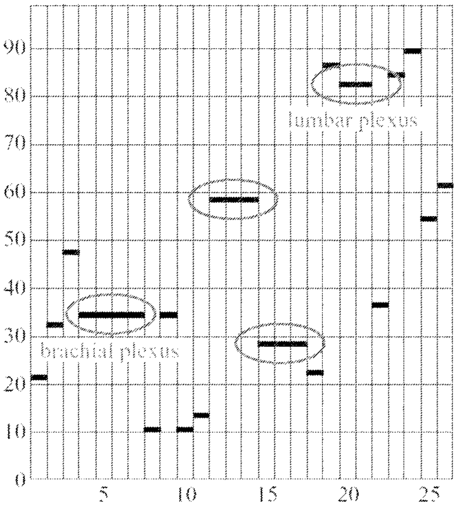

Fig. 7. The location of maximum column in each row of $\boldsymbol{P}$.

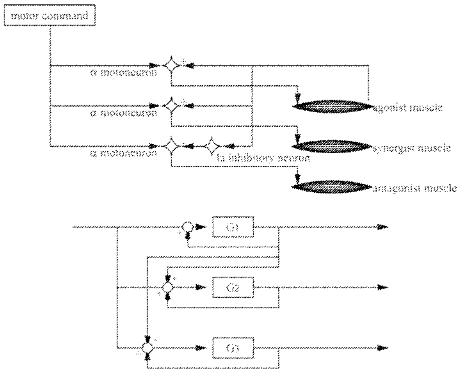

Fig. 8. The closed loop including the reflex (top) and corresponding neural network model (bottom).

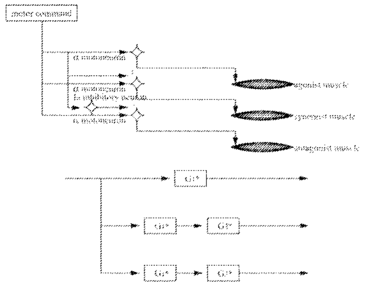

Fig. 9. The open loop model converted from the closed loop model (top) and corresponding neural network model (bottom).

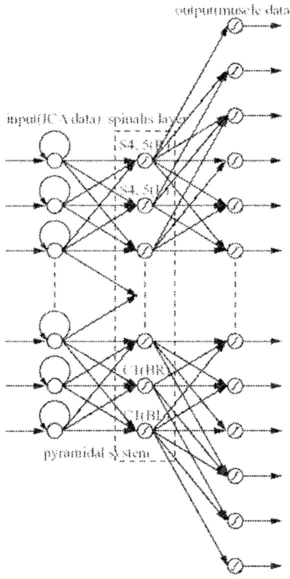

Fig. 10. The neural network model with three layers of the neuromuscular system.

individual wires. The connections between the neurons in the middle and output layers are determined according to the anatomical connection between the spinal rami and muscles.

IV. Experiments

A. Identification of the Neuromuscular Model Parameters

We trained the neural network of Fig. 10 so that it outputs the muscle tensions estimated for a walk motion with 338 frames when the inputs are the independent signals derived from ICA of the muscle tensions. The learning loop was repeated 200 times, which resulted in a

TABLE II

Weight Parameters of L2

muscle	weight	muscle	weight
psoas major	1.98E+00	iliacus	1.37E+00
sartorius	1.77E+00	rectus femoris	1.54E+00
vastus lateralis	2.21E+00	vastus medialis	1.55E+00
vastus intermedius	2.49E+00	gracilis	1.13E+00
pectineus	1.33E+00	adductor longus	1.48E+00
adductor brevis	2.65E+00	adductor magnus	2.48E+00
quadratus lumborum	1.76E+00		

TABLE III

Weight Parameters of L3

muscle	weight	muscle	weight
psoas major	1.56E+00	iliacus	1.98E+00
sartorius	2.05E+00	rectus femoris	2.47E+00
vastus lateralis	1.75E+00	vastus medialis	1.19E+00
vastus intermedius	2.27E+00	gracilis	1.12E+00
pectineus	1.12E+00	adductor longus	2.53E+00
adductor brevis	1.03E+00	adductor magnus	9.69E-01
obturator externus	2.49E+00	quadratus lumborum	1.32E+00

neural network with 2.08% maximum and 1.26% average errors.

Tables II to VI show the weight parameters obtained for the connections from selected neurons in the middle layer to their associated output neurons. These results show the following interesting properties:

- All the agonist muscles for hip flexon (iliacus, sartorius, rectus femoris, and pectineus) have negative values in Tables II and III.
- All the agonist muscles for knee extension (quadriceps femoris) have the same signs in each of Tables II–IV.
- All the agonist muscles for hip extension (gluteus maximus, biceps femoris, semitendinosus, and semimembranosus) have the same signs in each of Tables V and VI.
- The agonist muscles for dorsi flexion of the ankle joint (tibialis anterior, peroneus tertius, and extensor digitorum longus) and those for plantar flexion (gastrocnemius, soleus, and plantaris) have the opposite signs in Table VI.

TABLE IV

Weight Parameters of L4

muscle	weight	muscle	weight
psoas minor	-4.68E-01	rectus femoris	-2.82E-01
vastus lateralis	-5.62E-01	vastus medialis	-7.52E-01
vastus intermedius	8.87E-01	adductor longus	1.14E+00
adductor magnus	1.52E+00	gluteus medius	3.05E+00
gluteus minimus	5.90E-01	tensor fasciae latae	2.62E-01
obturator externus	1.65E+00	tibialis anterior	-6.49E-04
popliteus	4.19E-02	tibialis posterior	1.95E+00

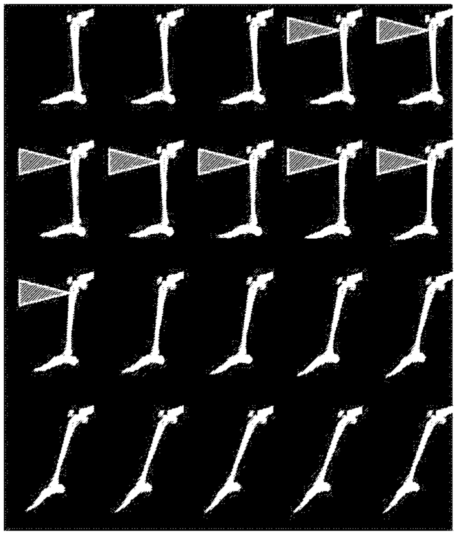

Fig. 11. Snapshots from the simulation of patellar tendon reflex. The red arrow is drawn while the unit signal to the nerve was applied.

B. Simulation of Patellar Tendon Reflex

As an application of the neuromuscular model, we simulated the patellar tendon reflex which is a typical example of stretch reflex. The algorithm described in Section II-C was used for the simulation. Patellar tendon reflex is a phenomena where the lower leg suddenly jerks forward when we tap just below the knee. This is explained by stretch reflex of the quadriceps due to the extension of its tendon.

In the simulation, we emulated the signal caused by the reflex by supplying a unit signal to L2–L4 where the quadriceps are connected. Fig. 11 shows the snapshots from the simulated motion. Note that the resulting motion is very similar to human patellar tendon reflex.

V. Conclusion

In this paper, we developed a macroscopic model of the human neuromuscular system and identified its parameters by muscle tensions estimated for several sequences of captured motions. We applied independent component analysis (ICA) and showed that the tensions of almost 1000 muscles can be represented by a linear combination of about 100 independent signals. The analysis of the mapping matrix ***W*** also implied that most of the independent signals can be regarded as nerve signals. Based on these observations, we built a neural network model with three layers and identified its parameters using experimental data. The weighting parameters showed interesting properties: weights of the agonist muscles connected to the same nerve have the same sign, while those of the antagonist ones have the opposite. Furthermore, the forward simulation based on

TABLE V

Weight Parameters of L5

muscle	weight	muscle	weight
gluteus maximus	-7.14E-02	gluteus medius	-8.03E-01
gluteus minimus	-5.78E-01	tensor fasciae latae	6.78E-01
obturator internus	-8.43E-02	gemellus superior	9.08E-01
gemellus inferior	3.86E-01	quadratus femoris	-4.27E-01
biceps femoris	-6.28E-01	semitendinosus	-2.69E-01
semimembranosus	-2.88E-01	tibialis anterior	1.52E-01
extensor hallucis longus	1.97E-01	extensor digitorum longus	1.67E+00
peroneus tertius	1.99E+00	popliteus	5.13E-01
flexor hallucis longus	-9.32E-01	flexor digitorum longus	-9.25E-01
tibialis posterior	3.42E-02	peroneus longus	1.89E-01
peroneus brevis	1.51E+00	lumbrical	4.09E-01

TABLE VI

Weight Parameters of S1

muscle	weight	muscle	weight
gluteus maximus	1.85E+00	gluteus medius	1.53E+00
gluteus minimus	1.97E+00	tensor fasciae latae	2.33E+00
piriformis	1.36E+00	obturator internus	7.22E-01
gemellus superior	1.22E+00	gemellus inferior	3.13E-01
quadratus femoris	1.02E+00	biceps femoris	1.48E+00
semitendinosus	4.81E-01	semimembranosus	8.83E-01
tibialis anterior	5.51E-01	extensor digitorum longus	-4.54E-01
peroneus tertius	5.45E-01	gastrocnemius	-1.97E-01
soleus	-7.58E-01	plantaris	-2.49E-01
popliteus	-4.63E-01	flexor hallucis longus	1.02E+00
flexor digitorum longus	7.67E-01	tibialis posterior	-7.19E-01
peroneus longus	8.17E-01	peroneus brevis	-7.40E-01
extensor digitorum brevis	-2.15E-01	abductor hallucis	-7.75E-01
flexor digiti minimi brevis	-6.24E-02	abductor digiti minimi	3.69E-01
lumbrical	5.32E-01	flexor hallucis brevis	-7.06E-01

the input signal emulating patellar reflex resulted in a motion very similar to actual knee jerk motion of the human.

Acknowledgement

The authors gratefully acknowledge the support by the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Scientific Research (S) 15100002.

References

[1] T. Flash and N. Hogan, "The coordination of arm movements: An experimentally confirmed mathematical model," The Journal of Neuroscience, vol. 5, pp. 1688–1703, 1985.

[2] M. Katato, Y. Maeda, Y. Uno, and R. Suzuki, "Trahectory formation of arm movement by cascade neural network model based on minimum torque-change criterion," Biological Cybernetics, vol. 62, no. 4, pp. 275 288, 1990.

[3] S. Delp and J. Loan, "A computational framework for simulating and analyzing human and animal movement," IEEE Computing in Science and Engineering, vol. 2, pp. 46–55, 2000.

[4] T. Komura, P. Prokopow, and A. Nagano, "Evaluation of the influence of muscle deactivation on other muscles and joints during gait motion," Jounral of Biomechanics, vol. 37, no. 4, pp. 425–436, 2004.

[5] F. Anderson and M. Pandy, "Static and dynamic optimization solutions for gait are practically equivalent," Journal of Biomechanics, vol. 34, pp. 153 161, 2001.

[6] Y. Nakamura, K. Yamane, Y. Fujita, and I. Suzuki, "Somatosensory computation for man-machine interface from motion capture data and musculoskeletal human model," IEEE Transactions on Robotics, vol. 21, no. 1, pp. 58 66, 2005.

[7] K. Yamane, Y. Fujita, and Y. Nakamura, "Estimation of physically and physiologically valid somatosensory information," in Proceedings of IEEE International Conference on Robotics and Automation, Barcelona,Spain, April 2005, pp. 2635–2641.

[8] A. Hill, "The heat of shortening and the dynamic constants of muscle," in Proceedings of the Royal Society of London, vol. B126, 1938, pp. 136 195.

[9] R. Featherstone, Robot Dynamics Algorithm. Boston, MA: Kluwer Academic Publishers, 1987.

[10] K. Yamane and Y. Nakamura, "Efficient Parallel Dynamics Computation of Human Figures," in Proceedings of the IEEE International Conference on Robotics and Automation, May 2002, pp. 530 537.

[11] A. Hyvärinen, J. Karhunen, and E. Oja, Independent Component Analysis. John Wiley & Sons, 2001.