Somatosensory Computation for Man-Machine Interface from Motion Capture Data and Musculoskeletal Human Model

Y. Nakamura, K. Yamane, Y. Fujita and I. Suzuki

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 discuss the computation of somatosensory information from motion capture data. The efficient computational algorithms previously developed by the authors for multibody systems such as humanoid robots are applied to a musculoskeletal model of the human body. The somatosensory information includes tension, length, and velocity of the muscles, tension of the tendons and ligaments, pressure of the cartilages, and stress of the bones. The technological development aims to define a higher dimensional man-machine interface and to open the door to the cognitive-level communication of humans and machines.

1 Introduction

The relationship between humans and machines will show qualitative differences through technical developments of communications and interactions. Such communications and interactions would imply not only explicit and conscious signal exchange but also implicit and subconscious one.

In this paper, we discuss the computation of somatosensory information from motion capture data. The efficient computational algorithms previously developed by the authors for multibody systems such as humanoid robots are applied to a high dimensional musculoskeletal model of the human body. The somatosensory information includes tension, length, and velocity of the muscles, tension of the tendons and ligaments, pressure of the cartilages, and stress of the bones.

The somatosensory information is the information sensed deep inside the human body and has not been put much emphasis on for applications other than medicine and rehabilitation. Motion capture systems have been used to capture the superficial change of human body. By combining the motion capture system, the musculoskeletal model of human body, and the efficient computation of kinematics and dynamics, the authors are exploring the possibility of realtime computation of the somatosensory information.

We know from our daily experiences that we subconsciously compute and estimate the somatosensory information of others. Such information is used in our cognitive process and helps us to understand the others' intention, mental states, and relationship with us. When the man-machine interface obtains access to the somatosensory information, machines make the first step to understand humans. Such a communication between humans and machines could be termed as a cognitive-level communication. The technology will make it possible to build such robots that naturally reach out humans in need.

A body of previous research on motion analysis and motion simulation of musculoskeletal human model has been conducted in the field of sports science and medicine [1] [2]. Their main purposes are to use human somatosensory data in order to improve motion patterns of athletes, and to monitor recovery of those who are in rehabilitation of mobility. These researches adopted a simplified model of the whole body or a rather detailed model limited to a small part of the body, due to the high computational cost of the detailed whole-body human model.

The computation of whole-body somatosensory information was focused in [3], [4], and [5], although the efficiency of computational algorithms was not paid much attention.

The authors developed an efficient algorithm for forward dynamics computation of structure-varying kinematic chains [6] to handle articulated multibody systems with the highest generality. The algorithm was recently improved to deliver order N efficiency with a single processor [7] and order $\log N$ efficiency with processors as many as order N [8] where N is the number of bodies.

A near realtime optical motion capture system was also developed by Kurihara et al.[9] in the authors' group and will be used to capture the target motions.

We constructed the musculoskeletal human model so that the dynamics computation algorithms for multibody systems can be applied straightforwardly as we explain in the following section in detail. Each of muscles, tendons, and ligaments is represented by a single wire which has two ends and via-points, generates its tension, and changes its length (only for muscles). The tension of a wire is assumed uniform throughout it and the via-points change the direction of wire without friction. Cartilages are modeled by linear springs.

The forward/inverse dynamics computation of musculoskeletal human model using the motion capture data allows us to compute somatosensory information. Applications of this technology include sports science, rehabilitation and medicine as it originally targeted. Being integrated with realtime technology and efficient algorithms, the technology will find it applications in the higher dimensional man-machine interface and make the foundation of cognitive-level communication between humans and machines.

2 Musculoskeletal Human Model

2.1 Elements and attributes

We designed the detailed human model as shown in Figure 1. It is comprised of the skeleton and the musculotendon network. The skeleton is a set of bones grouped into suitable number of body parts. The musculotendon network is a set of muscles, tendons, and ligaments spreading and straining among bones. Each attribute of these elements is modeled as follows:

(1) Bone : Rigid link element with mass and inertia.

(2) Muscle : Active constrictive wire actuator.

(3) Tendon : Passive constrictive wire that couples with a muscle and transmits its power.

(4) Ligament : Passive constrictive wire that connects bones and restrains its relative movement.

(5) Cartilage : Passive linear spring that connects bones and restrains its relative movement.

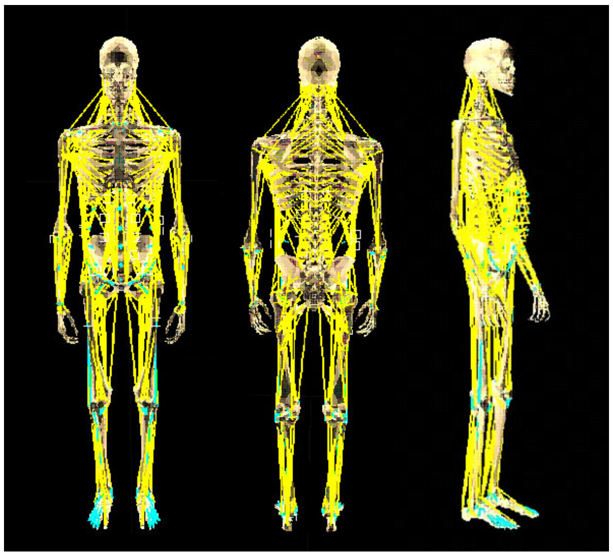

Figure 1: Musculoskeletal human model

2.2 Skeleton model

We used polygon geometric data of the bones commercially available for computer graphics [12]. We also produced polygon data by reconstructing the geometry from CT cross-sections. Both models are based on the standard body of European male.

In the beginning, for an experimentation of the whole body dynamics, we grouped bones under some simplification. In the concrete, the head and hands are counted as one link of the body respectively. The foot is modeled by two bodies of a toe and the other part of foot. Thus facial muscles and short musculotendons on hands and feet are mostly cut out. We applied these simplification and grouped about 200 bones into 53 links. We modeled every joint as a spherical joint with three DOF.

2.3 Musculotendon model

There are various types of muscles, tendons and ligaments. For instance, there are large muscles like triangular muscles, long muscles winding bones and muscles with furcation like biceps. They can be classified into the following patterns:

(1) one to be replaced by a simple wire that connect an origin and an end,

(2) one to be replaced by a wire that has an origin, via-points and an end,

(3) one to be replaced by several wires forming a fork at a virtual bone,

(4) one to be replaced by a set of simple wires,

(5) one to be replaced by a composite of those.

We describe typical examples of the above patterns successively.

2.3.1 simple wire model of muscles

Most of muscles, tendons, and ligaments are modeled as a wire which has only an origin and an end. Figure 2 is the model of M. Teres Minor, which ends at Humerus. In this way, we substituted one wire for a serial connection of a tendon, a muscle, and a tendon.

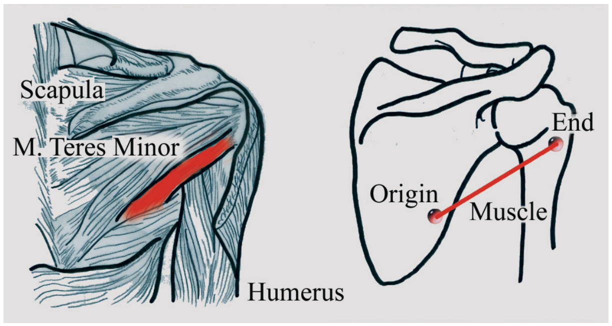

Figure 2: Model of a simple muscle

2.3.2 muscle model with via-points

There are muscles, tendons, and ligaments that wreathe around bones or cannot be modeled by simple lines connecting two points. We represent a curved path of a muscle, tendon, or ligament by segmented lines. Namely, we place via-points when a wire wreathes around a bone or when tendons are sheaved by a synovial sheath. A path of segmented lines has an origin, an end, and corners in the middle, which are called via-points. We assume that muscles, tendons, and ligaments are constrained to bones at via-points, where the muscles, tendons, and ligaments can slide without friction. Also assumed is that a muscle, tendon, or ligament has a uniform magnitude of tension throughout it and changes the direction of tension at via-points. An example is seen in Figure 3.

2.3.3 muscle model with virtual links

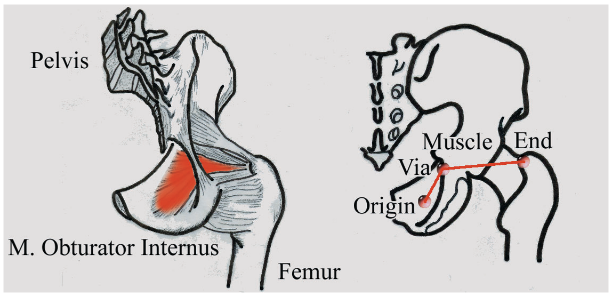

Figure 3: Muscle with via-points

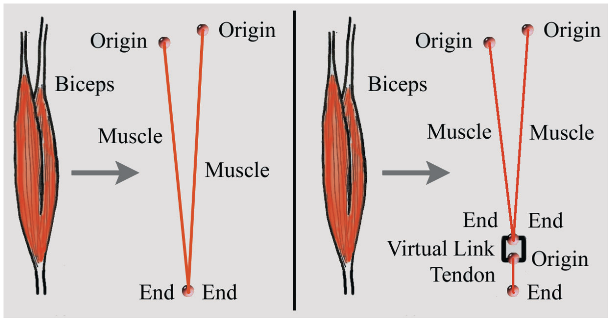

Figure 4: Two models of furcation

Muscles, tendons, and ligaments are not always connected in series. They sometimes show furcation. Figure 4 shows two simple cases with furcation. Although the both cases are equivalent, the right hand case adopts a virtual link, which allows more flexibility in representing muscle, tendon, and ligament networks. A virtual link is modeled as a small link without mass. Figure 5 is the model of M. Biceps Brachii which has a bifurcation and a via-point.

At the last part of M. Biceps Brachii, two muscles are connected to a tendon. The tendon separates into two branches, each of which inserts to a respective bone. Therefore, the furcation model of the tendon is necessary to render the function of M. Biceps Brachii. Virtual links are needed to model furcations according to the fundamental rule that a wire originates from a point, passes through zero or more via-points, and ends at a point. A virtual link is defined as a link without mass and used to translate tensions of wires.

2.3.4 muscle model comprised of several wires

We modeled some wide-spanned muscles using several simple wires. A typical example of such is the modeling of M. Pectoralis Major as shown in Figure 6.

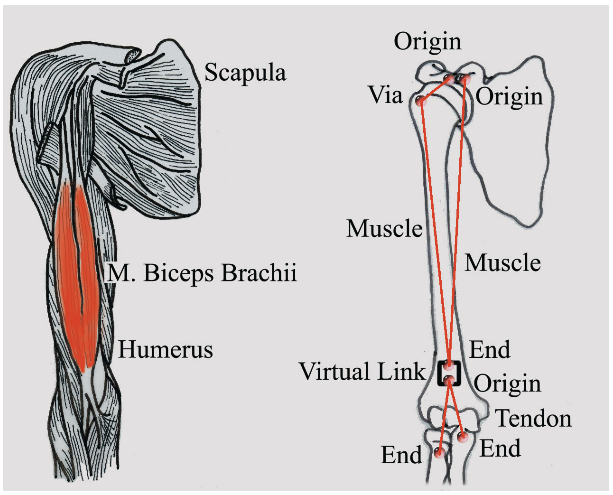

Figure 5: Model of M. Biceps Brachii

2.3.5 composite model

There are complex parts that require for modeling to combine the above mentioned models. Figure 7 is the example of the ligaments on elbow. Anular Ligament of Radius forms a ring, allows independent pronation and supination of radius, and constrains radius and ulna to interlock at flexion and extension of elbow. We used a virtual link and many via-points to model its function as seen in Figure 7.

Consulting an anatomy reference [13], the model of musculotendon network was built. The model covers the most parts of the musculotendon networks of body except for those in the head, hands, and

The dimensions of model is summarized in Table 1. The bones of 200 pieces are gouped into 53 bones, most of which are connected to the neighboring bones with 3DOF spherical joints. The exceptions are to cover complex mobility due to constraints such as surface contacts and closed loops. The skeletal system has 155DOF. In addition, we introduce 28 virtual links, which have $28 \times 6 = 168$ DOF. In total, the DOF of links becomes 323. For modeling muscles, tendons, ligaments, and cartilages, we used 366, 56, 91, and 34 wires respectively. Accordingly, we have 547 wires in total.

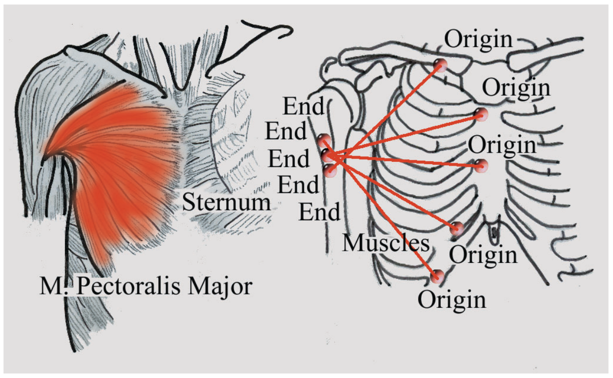

Figure 6: Model of M. Pectoralis Major

bones	200
groups of bones[*1]	53
virtual links[*2] (n_{VL})	28
skeletal DOF (n_{sk})	155
Total DOF (N_G)	323
wires for muscles (N_m)	366
wires for tendons	56
wires for ligaments	91
wires for cartilages	34
Total number of wires (N_l)	547

Table 1: Dimensions of the musculoskeletal model. The bones of 53 groups[*1] consist of two 6DOF bones (hip and chest), two 1DOF bones (toes), two zero-DOF bones (knee-caps), and forty-seven 3DOF bones (the others). Virtual links[*2] are modeld with 6DOF each.

3 Dynamics Computation of Musculoskeletal Model

3.1 Forward dynamics

Forward dynamics means computation to estimate motion, namely accelerations, of the generalized coordinates for the given generalized forces. In this paper, forward dynamics is to simulate the whole body motion from the given wire tensions of muscles for the musculoskeletal human model.

The computational algorithm to be presented in this subsection is based on the method proposed in [6] which was originally developed for general multibody systems including closed kinematic chains and even for those changing the structure in time. The computation includes the following three steps:

STEP 1 Compute $\boldsymbol{J} \in \boldsymbol{R}^{N_l \times N_G}$, the Jacobian matrix of wire and spring lengths with

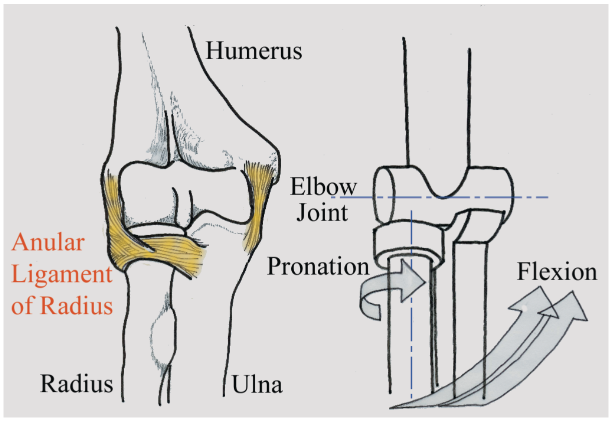

Figure 7: Mechanism of elbow joint

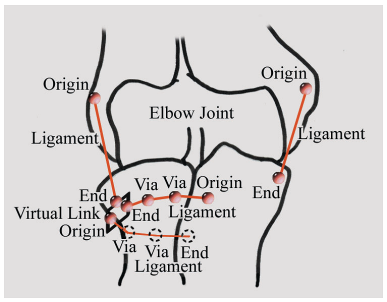

Figure 8: Model of ligaments on elbow joint

respect to the generalized coordinates defined as

$$\boldsymbol{J} = \frac{\partial \boldsymbol{l}}{\partial \boldsymbol{\theta}_G} \tag{1}$$

where $\boldsymbol{l} \in \boldsymbol{R}^{N_l}$ is the lengths of wires and springs, $\boldsymbol{\theta}_G \in \boldsymbol{R}^{N_G}$ is the generalized coordinate, and N_l and N_G are the number of wires and springs and the degrees of freedom (DOF) of the model respectively. Namely, $N_l = 547$ and $N_G = 323$. The number of wires for muscles is also represented by N_m in Table 1.

STEP 2 Map the given wire tensions into the generalized force using the Jacobian matrix obtained above. In the skeletal model we adopted, the generalized force consists of three dimensional moments of the spherical joints.

STEP 3 Solve for the accelerations of the generalized coordinates from the generalized force and integrate them to obtain the whole body motion.

For the definitions of the generalized coordinates and the generalized force, and their determination for complex constrained multibody systems, please refer to [6]. In the following paragraphs, we provide more precise description of the procedures.

3.1.1 [STEP 1] computation of the Jacobian matrix

In order to compute the Jacobian matrix of Equation (1), we first consider $\boldsymbol{J}_{Li}$, the Jacobian matrix of the length of the i-th wire with respect to the generalized coordinates. The matrix satisfies the following relationship:

$$\dot{l}_i = \boldsymbol{J}_{Li}\dot{\boldsymbol{\theta}}_G \tag{2}$$

where l_i is the length of the i-th wire.

We suppose that the i-th wire is comprised of m_i ($m_i \geq 0$) via-points plus the initial and final points. Let the length between the j-th and $(j+1)$-th via-points be $l_{i,j}$. Also let the Jacobian matrix of $l_{i,j}$ with respect to the generalized coordinates be $\boldsymbol{J}_{Li,j}$. Then $\boldsymbol{J}_{Li}$ can be represented by the sum of $\boldsymbol{J}_{Li,j}$ as follows:

$$\boldsymbol{J}_{Li} = \sum_{j=0}^{m_i-1} \boldsymbol{J}_{Li,j}. \tag{3}$$

Using $\boldsymbol{p}_{i,j}$, the position of the j-th via-point of the i-th wire, $l_{i,j}$ can be described as follows:

$$l_{i,j}^2 = \left(\boldsymbol{p}_{i,j+1} - \boldsymbol{p}_{i,j}\right)^T \left(\boldsymbol{p}_{i,j+1} - \boldsymbol{p}_{i,j}\right) \tag{4}$$

Differentiating Equation (4) with respect to the generalized coordinates yields the following equation:

$$\begin{aligned} \boldsymbol{J}_{Li,j} = \frac{\partial l_{i,j}}{\partial \boldsymbol{\theta}_G} &= \frac{1}{l_{i,j}} \left(\boldsymbol{p}_{i,j+1} - \boldsymbol{p}_{i,j}\right)^T \frac{\partial}{\partial \boldsymbol{\theta}_G} \left(\boldsymbol{p}_{i,j+1} - \boldsymbol{p}_{i,j}\right) && (5) \\ &= \frac{1}{l_{i,j}} \left(\boldsymbol{p}_{i,j+1} - \boldsymbol{p}_{i,j}\right)^T \left(\boldsymbol{J}_{p_{i,j+1}} - \boldsymbol{J}_{p_{i,j}}\right) && (6) \end{aligned}$$

where $\boldsymbol{J}_{p_{i,j}} = \partial \boldsymbol{p}_{i,j} / \partial \boldsymbol{\theta}_G$ implies the Jacobian matrix of $\boldsymbol{p}_{i,j}$ with respect to the generalized coordinates and can be calculated according to the standard procedure [14]. From Equation (3) and Equation (6) we can compute $\boldsymbol{J}_{Li}$. As a result, we obtain $\boldsymbol{J}$, the Jacobian matrix of Equation (1), by placing $\boldsymbol{J}_{Li}$ of the i-th wire in the i-th row of $\boldsymbol{J}$.

3.1.2 [STEP 2] mapping wire tensions into joint torques

We compute the joint torques $\boldsymbol{\tau}_G \in \boldsymbol{R}^{N_G}$ from the given wire and spring tensions $\boldsymbol{f}$ of muscles, tendons, ligaments, and cartilages. The mapping is linear and represented by the following equation:

$$\boldsymbol{\tau}_G = \boldsymbol{J}^T \boldsymbol{f}. \tag{7}$$

3.1.3 [STEP 3] solving accelerations

Without losing the generality of [6] we improved the efficiency of computation in [7] [8]. We use these algorithms to solve accelerations of the generalized coordinates.

For an articulated multibody system consisting of N bodies, the computational complexity of the algorithm is $O(N)$. If the algorithm is implemented on a parallel processing system with $O(N)$ processors, the computational time is reduced to $O(\log N)$.

The same algorithm is used for parallel computation on any number of processors for higher efficiency and even for serial computation on a single processor as well without modification.

3.2 Inverse dynamics with force distribution

Inverse dynamics computation is to compute the tensile strengths of muscles, tendons, and ligaments and spring forces of cartilages from the motion data obtained through, for example, motion capture. The computation requires the following procedures:

STEP 1 Apply the standard inverse dynamics computation to the given motion data, and compute the joint torques. We can treat the mechanism as a serial tree-structure kinematic chain with the fixed base, assuming that the chain is cut open if it has loops and also assuming that free joints, if any, are actively driven. There assumptions are considered appropriately and used to compute the wire tensions. This is following the basic idea of [6].

STEP 2 Map the joint torques to the wire tensions $\boldsymbol{f}$. For this computation we first determine the generalized coordinates that may vary in time due to the change of constraints. We use $\vec{J}$ the Jacobian matrix of the wires with respect to the generalized coordinates and $\vec{J_\theta}$ the Jacobian matrix of all the joints of the serial tree-structure kinematic chain with respect to the generalized coordinates for the mapping [6]. The mapping problem is not straightforward because the number of wires are much larger than the DOF of the whole system. We used mathematical programming methods to solve this problem of redundancy.

3.2.1 optimization of tensions

The relationship between the generalized forces $\boldsymbol{\tau}_G$ and the wire tensions $\boldsymbol{f}$ is represented in Equation (7). The relationship between the joint torques $\boldsymbol{\tau}_\theta$ and the generalized forces $\boldsymbol{\tau}_G$ is on the other hand provided by the following equation:

$$\boldsymbol{\tau}_G = \boldsymbol{J}_\theta^T \boldsymbol{\tau}_\theta. \qquad (8)$$

The generalized forces are computed by simply substituting the joint torques computed in STEP1 into the above equation. Accordingly, the problem remains in solving Equation (7) for $\boldsymbol{f}$ using the obtained generalized force $\boldsymbol{\tau}_G$. Since the number of wires (547) is larger than the total DOF (323) that is even greater than the DOF of the generalized coordinates, solving $\boldsymbol{f}$ from $\boldsymbol{\tau}_G$ is a highly redundant problem.

The wire tensions of muscles, tendons, and ligaments only work for contractive direction, while spring tensions of cartilages work for both contactive and expansive directions. The characteristics are represented by the following constraint condition:

$$\boldsymbol{E}_{mtl}\boldsymbol{f} \leq \boldsymbol{0}. \tag{9}$$

where $\boldsymbol{E}_{mtl}$ is a matrix extracting tensions of muscles, tendons, and ligaments from $\boldsymbol{f}$. It consists of unit row vectors as many as number of muscles, tendons, and ligaments. A row vector has one at the position of corresponding muscle, tendon, or ligament.

Technically it is a problem of actuation redundancy [10] subject to an equality condition of Equation (7) and an inequality condition of Equation (9).

The goal of inverse dynamics computation of musculoskeletal human model is to estimate the activity of muscles. To meet the goal we have to carefully set the performance index for the actuation redundancy problem. We will have to develop the knowledge of how humans use the redundancy and to formulate the performance index to incorporate the knowledge.

In order to proceed the discussion of inverse dynamics computation, we now assume that such knowledge is represented in the following form of nominal use of muscles:

$$\boldsymbol{f}_s = \boldsymbol{f}_s(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}, \boldsymbol{s}, t) \tag{10}$$

where $\boldsymbol{\theta}$ is the joint variable. Vector $\boldsymbol{s}$ implies sensory information of all kind, and t represents time. Since a bundle of muscle fibers of skeletal muscles has muscle spindles as stretch receptors, the information of muscle lengths and, therefore, of joint variables are included in $\boldsymbol{s}$. In the above equation, we include $\boldsymbol{\theta}$ and $\dot{\boldsymbol{\theta}}$ to make the functional dependence of $\boldsymbol{f}$ to them. Our problem now can be represented by the following expression:

Find $\boldsymbol{f}$ *that minimizes*

$$Z = \frac{1}{2}|\ \boldsymbol{f} - \boldsymbol{f}_s(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}, \boldsymbol{s}, t)\ |^2 \tag{11}$$

subject to the following constraints:

$$\boldsymbol{\tau}_G = \boldsymbol{J}^T\boldsymbol{f} \tag{12}$$

$$\boldsymbol{E}_{mtl}\boldsymbol{f} \leq \boldsymbol{0}. \tag{13}$$

3.2.2 solving with quadratic programming

Solving the problem of Equation (11) through Equation (13) forms a quadratic programming (QP) problem, which is commonly represented as the following optimization:

Find an extreme value of the objective function,

$$Z = \frac{1}{2}\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{c}^T\boldsymbol{x} \tag{14}$$

under linear equality and inequality constraints for $\boldsymbol{x}$.

If the matrix $\boldsymbol{Q}$ is a symmetric positive semidefinite matrix, the problem is called a convex quadratic programming problem, for which efficient algorithms are known. For our case, $\boldsymbol{Q}$ is an identity matrix from Equation (11) and, therefore, positive definite. We can introduce weighting ratios among the components of $\boldsymbol{f}$, when the matrix becomes diagonal with positive entries and remains positive definite. Note that $\boldsymbol{c} = \boldsymbol{f}_s(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}, \boldsymbol{s}, t)$ in Equation (11). Also note that the third term, $\frac{1}{2}\boldsymbol{f}_s{}^T\boldsymbol{f}_s$, in Equation (11) is independent to $\boldsymbol{f}$ and, therefore, can be disregarded for optimization.

3.2.3 solving with linear programming

Although quadratic programming has efficient algorithms for computation, they are commonly computationally expensive. When we need to pursue computational speed at an acceptable cost of computational accuracy, we can apply linear programming as an approximating solver of the original problem.

By defining

$$\Delta\boldsymbol{f} = \boldsymbol{f} - \boldsymbol{f}_s(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}, \boldsymbol{s}, t) \tag{15}$$

$$\Delta\boldsymbol{\tau}_G = \boldsymbol{\tau}_G - \boldsymbol{J}^T\boldsymbol{f}_s(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}, \boldsymbol{s}, t) \tag{16}$$

we can equivalently reformulate the problem of Equation (11) through Equation (13) as follows:

Find $\Delta\boldsymbol{f}$ *that minimizes*

$$Z = \frac{1}{2}|\ \Delta\boldsymbol{f}\ |^2 \tag{17}$$

subject to the following constraints:

$$\Delta\boldsymbol{\tau}_G = \boldsymbol{J}^T\Delta\boldsymbol{f} \tag{18}$$

$$\boldsymbol{E}_{mtl}\{\Delta\boldsymbol{f} + \boldsymbol{f}_s(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}, \boldsymbol{s}, t)\} \le \boldsymbol{0}. \tag{19}$$

The above fromulation remains a quadratic programming problem. We approximately simplify it into a linear programming problem by introducing variable vector $\boldsymbol{\delta}_f \in \boldsymbol{R}^{N_f}$ as follows:

For constant vector $\boldsymbol{a}$ *with positive components, find* $\boldsymbol{\delta}_f$ *and* $\Delta\boldsymbol{f}$ *that minimize*

$$Z = \boldsymbol{a}^T\boldsymbol{\delta}_f \tag{20}$$

subject to the following constraints:

$$\Delta\boldsymbol{\tau}_G = \boldsymbol{J}^T\Delta\boldsymbol{f} \tag{21}$$

$$\boldsymbol{E}_{mtl}\{\Delta\boldsymbol{f} + \boldsymbol{f}_s(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}, \boldsymbol{s}, t)\} \le \boldsymbol{0} \tag{22}$$

$$-\boldsymbol{\delta}_f \leq \Delta \boldsymbol{f} \leq \boldsymbol{\delta}_f \tag{23}$$

$$\boldsymbol{\delta}_f \geq \boldsymbol{0}. \tag{24}$$

Vector $\boldsymbol{a}$ typically has one in every component, which approximates the case of Equation (17), while $\boldsymbol{a}$ with uneven positive components approximates a quadratic performance index with the weighted norm.

As the constraint conditions of Equation (21) through Equation (24) and the performance index of Equation (20) are all linear in the above approximated formulation, we can apply linear programming (LP) to the problem. The simplex method is known as an efficient algorithm for LP.

4 Examples of Computation

4.1 Forward dynamics

Given the muscle tensions, forward dynamics computes the motion of the whole body. We modeled the dynamics of muscle wires by soft spring and damper located in parallel, while tendons, ligaments, and cartilages are modeled by hard spring and damper. A muscle wire tension controller was also designed as a feedback controller to compute the input torque. However, it was not easy to a practical controller to produce a relevant example motion by forward dynamics computation.

The example we adopted here is one for preliminary testing, which simulates a free gravity motion. We set all the muscle tensions zero. We fixed head position for the boundary condition. We see in Figure 9 the whole body gradually stretched down due to the gravity.

The simulation was executed on a Pentium IV 2 GHz machine, and the computational time was 0.5 [sec] for each step.

4.2 Inverse dynamics

In this subsection we show the results of inverse dynamics computation. In inverse dynamics, as we saw in the previous section, determination of $\boldsymbol{f}_s$ is fundamental and essential. It requires extensive experimental research to build the database of how the human uses the muscles. Such database will be of extreme importance for further research of somatosensory computation. We plan to continue the research and carry out experiments for accumulating the data in our future work.

In order to focus ourselves to the computational problems in the present paper, we assume $\boldsymbol{f}_s = 0$ and apply LP and QP.

We obtained the motion captured data of "kicking motion" using a motion capture system developed in our laboratory [9]. The data was for 7.29 [sec] of 143 frames captured at every 30 [msec]. We computed the inverse dynamics of the captured kicking motion to be reported below.

We have 155DOF for the skeletal system and 323DOF including virtual links. On the other hand, we have 366 muscle wires for driving the whole system. Through computation of the example, however, we learned that the number of driving inputs was not redundant and not even sufficient for producing arbitrary whole body motions. It became clear from

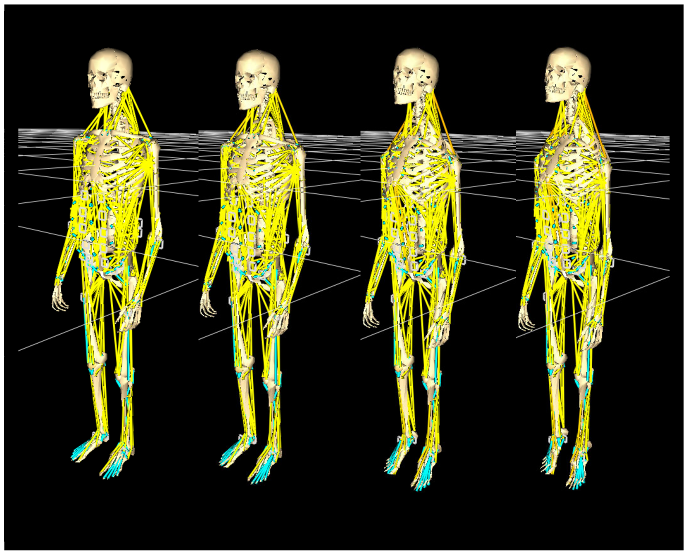

Figure 9: An Example of Forward Dynamics

the fact that there was no solution of $\boldsymbol{f}$ for the generalized force of $\boldsymbol{\tau_G}$ computed from the motion capture data of kicking motion. This was particularily seen for the joint torques of vertebras in the neck. Since there are 7 bones with a 3DOF spherical joint each and not many muscle wires in the neck, the joint torques are not fully represented by the wire tensions.

The results suggest that the forces acting to the whole body cannot be approximated only by the simple wire model of major muscles. The distributed forces of tissue and organs are passively generated in the human body by the skeletal muscles, and support the skeletal system. For the human dynamics model and the inverse dynamics computation based on the model, it would be necessary to build a more precise model taking account of the effects of tissue and organs.

To avoid the essential problem in the current computation, we introdused another veriable vector $\boldsymbol{\delta}_\tau \in \boldsymbol{R}^{N_G}$ and replaced the equality constraint of Equation (21) with an inequality constraint as follows:

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

$$Z = {\boldsymbol{a}_f}^T \boldsymbol{\delta}_f + {\boldsymbol{a}_\tau}^T \boldsymbol{\delta}_\tau. \tag{25}$$

subject to the following constraints:

$$-\boldsymbol{\delta}_\tau \leq \Delta\boldsymbol{\tau}_G - \boldsymbol{J}^T \Delta\boldsymbol{f} \leq \boldsymbol{\delta}_\tau \tag{26}$$

$$\boldsymbol{\delta}_\tau \geq \boldsymbol{0} \tag{27}$$

$$\boldsymbol{E}_{mtl}\{\Delta\boldsymbol{f} + \boldsymbol{f}_s(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}, \boldsymbol{s}, t)\} \leq \boldsymbol{0} \tag{28}$$

$$-\boldsymbol{\delta}_f \leq \Delta \boldsymbol{f} \leq \boldsymbol{\delta}_f \tag{29}$$

$$\boldsymbol{\delta}_f \geq \boldsymbol{0}. \tag{30}$$

This replacement is to relax the problem and to assure the existence of solution by allowing computational errors, where $\boldsymbol{\delta}_\tau$ represents the permissible range of error in the joint torques. The permissible error is intuitively selected by setting the ratio of the magnitudes of $\boldsymbol{a}_f$ and $\boldsymbol{a}_\tau$. The modified LP problem of Equation (25) through Equation (30) was implemented for computation and tested. Each of $\boldsymbol{a}_f$ and $\boldsymbol{a}_\tau$ was set to have uniform components. The ratio of their magnitudes, $\mid \boldsymbol{a}_\tau \mid / \mid \boldsymbol{a}_f \mid$, is represented by w_{LP}.

For comparison, we also modified the QP problem of Equation (11) through Equation (13) as follows:

Find $\boldsymbol{f}$ *that minimizes*

$$Z = \mid \boldsymbol{f} - \boldsymbol{f}_s(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}, \boldsymbol{s}, t) \mid ^2 + w_{QP} \mid \boldsymbol{\tau}_G - \boldsymbol{J}^T \boldsymbol{f} \mid ^2 \tag{31}$$

subject to the following constraints:

$$\boldsymbol{E}_{mtl} \boldsymbol{f} \leq \boldsymbol{0}. \tag{32}$$

where w_{QP} indicates the weighting parameter.

Note that the above QP problem has 547 variables to optimize, which is the dimension of $\boldsymbol{f}$. On the other hand, the LP problem of Equation (25) through Equation (30) has 1417 variables of $\Delta \boldsymbol{f}$, $\boldsymbol{\delta}_f$, and $\boldsymbol{\delta}_\tau$ to optimize.

For the computation of convex QP problems, we used the primal-dual interior-point algorithm routines of the free object-oriented software package for solving convex QP problem (OOQP[16]). For the computation of LP problems, we used the simplex method routines of the GNU free LP library (GLPK[15]). All the computation was executed on a PC with a single processor of Pentium IV, 2GHz.

The results of computation were summerized by the three graphs of Figure 10 through Figure 12. The weighting parameters were set $w_{LP} = 10^8$ and $w_{QP} = 10^6$. Since the performance index for QP is of squared magnitudes and that for LP is of linear magnitudes, the two weighting parameters cannot be simply compared. Note that vectors $\boldsymbol{\tau}_G$ and $\boldsymbol{f}$ were described in [Nm] and [N] respectively for numerical evaluation.

Figure 10 shows that LP takes 2.98 [sec] for each frame on average while QP takes 14.2 [sec]. LP is 4.77 times faster in computation than QP in spite of the fact that LP has 2.59 times more variables than QP.

Figure 11 represents $\mid \boldsymbol{\tau}_G - \boldsymbol{J}^T \boldsymbol{f} \mid ^2$ which is the squared magnitude of the error of constraints. Although the constraints were to be represented as equality ones in physical reality, they were replaced by inequality ones for computational stability. The error becomes very large for LP near the 50th and 110th frames, the latter of which corresponds to the end of the kick. Further increase of w_{LP} could not reduce the error of LP, while increase of w_{QP} could further reduce that of QP.

On ther other hand, Figure 12 shows $\mid \boldsymbol{f} - \boldsymbol{f}_s \mid ^2$ for both LP and QP. Since $\boldsymbol{f}_s = \boldsymbol{0}$ was assumed, the two curves imply the squared magnitudes of $\boldsymbol{f}$. Although increase of w_{QP} tended to reduce the error of equality constraints as mentioned above, it admitted increase of $\boldsymbol{f}$. The large fluctuation of the QP curve is due to the reratively smaller weight for $|\ \boldsymbol{f} - \boldsymbol{f}_s\ |^2$. In other words, the fluctuation can be reduced at the cost of the error of equality constraints.

Although it is a suggestion only from this particular example, the three graphs and the comparison of computational time mildly suggest that the LP problem described in Equation (25) through Equation (30) is an efficient formulation of the inverse dynamics problem in comparison with the QP problem.

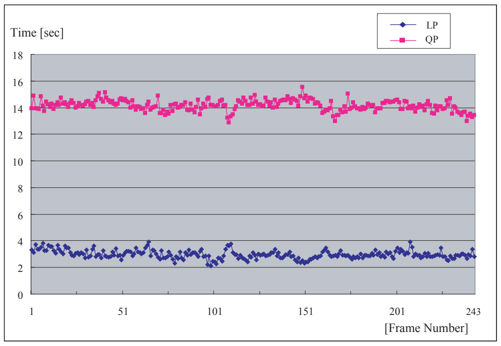

Figure 10: Comparison of computational time between LP and QP for inverse dynamics

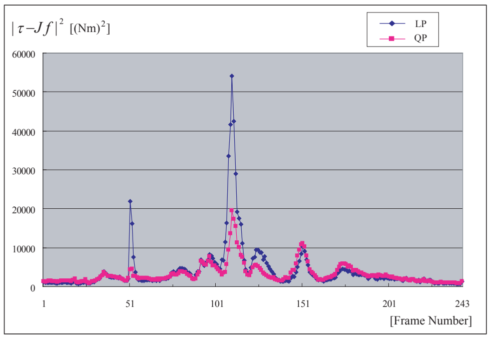

Figure 11: Comparison of $|\ \boldsymbol{\tau}_G - \boldsymbol{J}^T\boldsymbol{f}\ |^2$ between LP and QP for inverse dynamics

The result of LP computation was visualized as seen in Figure 13, where the colors of muscle wires change from yellow to red (from white to gray) as their tensions grow. The colors of other passive wires for tendons and ligaments also change from light blue to dark blue (from white to gray) as their tensions increase.

In the inverse dynamics computation, we have determined the muscular force by utilizing mathematical optimization methods. We have not started yet the physiological study of making the database on how we use our muscle tensions. The computational methods need to be extended to integrate the physiological knowledge, since the computed forces without such

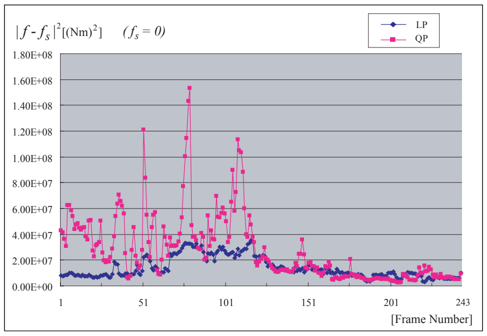

Figure 12: Comparison of $|\, \boldsymbol{f} \,|^2$ between LP and QP for inverse dynamics

knowledge lack the consistency with the actual human muscular forces. The contribution of this paper is the formulation and implementation of mathematical method that will provides somatosensory computation being combined with the physiological knowledge.

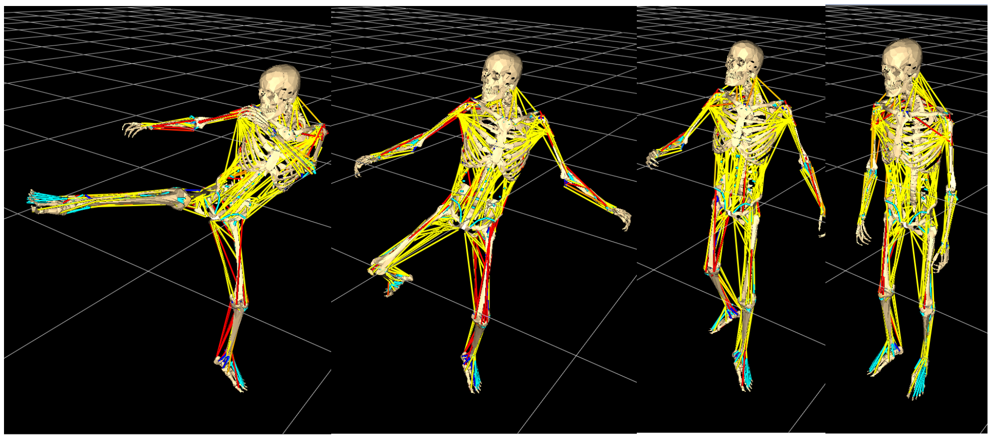

Figure 13: an Example of Inverse Dynamics

5 Conclusions

We proposed a method for modeling the human musculoskeletal dynamics, where the dynamics computation algorithms developed for kinematic chains are extended to compute the muscle and tendon tensions. We constructed a detailed human model based on this modeling method.

We developed computational algorithms for inverse and forward dynamics of wire-based human model on the basis of efficient multibody dynamics computation algorithms [7][8].

The computational algorithms were successfully implemented and used for numerical experiments to show their performance.

The computational problem of inverse dynamics is reduced to an optimization problem. The problem was formulated in two forms, namely, as a quadratic programming problem and as a linear programming problem. The two formulations were compared from the viewpoint of computational time, accuracy, and stability of solution. The results of example showed that the linear programming was efficient, accurate, and stable.

The optimization criteria need to be specifically and appropriately chosen to simulate the way how the human uses the muscle redundancy. We are currently searching for the criteria and modeling it. Concurrent use of electromyograph at motion capturing would be a promising approach.

The forward and inverse dynamics computations of the musculoskeletal human model using motion capture data allow us to compute the somatosensory information. Applications of this technology include sports science, rehabilitation and medicine as it was originally targeted. Integrated with realtime technology and efficient algorithms, the technology will find its applications in the higher dimensional man-machine interface and serve as the foundation of cognitive-level communication between humans and machines.

Acknowledgments

This research was supported by the CREST program, the Japan Science and Technology Corporation, and the Information-Technology Promotion Agency (IPA) of Japan.

References

[1] M. Kaya, H. Minamitani, K. Hase, and N. Yamazaki: Motion analysis of optimal rowing form by using biomechanical model, Proc. IEEE Engineering in Medicine and Biology, pp. 715-716, 1995.

[2] Taku Komura, Yoshihisa Shinagawa, and Tosiyasu L. Kunii: Creating and retargetting motion by the musculoskeletal human body model, The Visual Computer, Vol.16, No.5, pp.254-270, 2000.

[3] S.L. Delp and J.P. Loan: A software system to develop and analyze models of musculoskeletal structures, Computers in Biology and Medicine, vol. 25, pp. 21-34, 1995.

[4] S.L. Delp, and J.P. Loan: A computational framework for simulating and analyzing human and animal movement, IEEE Computing in Science and Engineering, vol. 2, pp. 46-55, 2000.

[5] Naoki Suzuki and Akihiro Takatsu: 3D and 4D visualization of morphological and functional information from the human body using noninvasive measurement data, Atlas of Visualization, Vol. 3 1997.

[6] Yoshihiko Nakamura and Katsu Yamane: Dynamics computation of structure-varying kinematic chains and its application to human figures, IEEE Transactions on Robotics and Automation, Vol.16, No.2, pp.124-134, 2000.

[7] Katsu Yamane and Yoshihiko Nakamura: O(N) forward dynamics computation of open kinematic chains based on the principle of virtual work, Proc. of IEEE International Conference on Robotics and Automation, Vol.3, pp.2824-2831, Seoul, Korea, May, 2001.

[8] Katsu Yamane and Yoshihiko Nakamura: Efficient parallel dynamics computation of human Figures, Proc. of IEEE International Conference on Robotics and Automation, Vol.1, pp.530-537, Washington D.C., U.S.A., May, 2002.

[9] K. Kurihara, S. Hoshino, K. Yamane and Y. Nakamura: Optical motion capture system with pan-tilt camera tracking and realtime data processing, Proc. of IEEE International Conference on Robotics and Automation, Vol.2, pp.1241-1248, Washington D.C., U.S.A., May, 2002.

[10] Yoshihiko Nakamura: Advanced Robotics: Redundancy and Optimization, Addison-Wesley, 1991.

[11] Francis D. Carlson and Douglas R. Wilkie: Muscle Physiology, Prentice-Hall Biological Science Series, 1974.

[12] The Viewpoint Medical Products: Viewpoint Corporation, http://www.viewpoint.com/, 2002.

[13] W. Kahle, H. Leonhardt, and W. Platzer: Taschenatlas der Anatomie, Band 1: Bewegungsapparat, Georg Thieme Verlag Stuttgart New York, Deutscher Taschenbuch Verlag, 1986.

[14] D.E. Orin and W.W. Schrader: Efficient computation of the Jacobian for robot manipulators, International Journal of Robotics Research, vol.3, no.4, pp.66-75, 1984.

[15] GLPK: http://www.gnu.org/software/glpk/glpk. M. html

[16] OOQP: http://www.cs.wisc.edu/∼swright/ooqp/