Proceedings of the 2nd International Symposium
on Adaptive Motion of Animals and Machines,
Kyoto, March.4-8, 2003

Dynamics Computation of Musculo-Skeletal Human Model Based on Efficient Algorithm for Closed Kinematic Chains

Yoshihiko Nakamura[*1], Katsu Yamane[*2], Ichiro Suzuki[*3] and Yusuke Fujita[*4]

[*1]Univ. of Tokyo., Dept. of Mechano-Informatics, Bunkyo, Tokyo 113–8656 Japan, nakamura@ynl.t.u-tokyo.ac.jp

[*2]Carnegie Mellon Univ., Robotic Institute, Pittsburgh PA 15213-3890 USA, kyamane@cs.cmu.edu

[*3]Univ. of Tokyo., Dept. of Mechano-Informatics, Bunkyo, Tokyo 113–8656 Japan, ichiro@ynl.t.u-tokyo.ac.jp

[*4]Univ. of Tokyo., Dept. of Mechano-Informatics, Bunkyo, Tokyo 113–8656 Japan, fujita@ynl.t.u-tokyo.ac.jp

Abstract

We have developed an efficient algorithm for dynamics computation of closed kinematic chains in the field of robotics. We applied it to far more complex Musculo-Skeletal human model we designed, and realized forward/inverse dynamcis computation of the model in a practical time. In this paper we described the methods to design Musculo-Skeletal human model using closed chain of links and wires, in order to apply the efficient dynamics computation. We also described dynamics computation method to compute somatosensory information of Musculo-Skeletal human model based on efficient algorithm for closed kinematic chains.

1. Introduction

There are many researches on motion analysis and motion simulations of Musculo-Skeletal human model in the field of sports science and medicine[1][1][3]. The purpose of these researches are to analyze human somatosensory data and use them to improve motion styles of atheletes, to evaluate motions of the handicapped, and to support rehabilitation plannings. However, these researches uses simplified models or the models they use are limited to small part of the body, so they could not deal with whole body motions. This was due to the high computational cost of whole body detailed human model.

Meanwhile, we have developed an efficient algorithm for dynamics computation of closed kinematic chains[8] in the field of robotics. We applied it to the Musculo-Skeletal human model we designed, and realized forward/inverse dynamics computation of the model in a practical time. The model and dynamics computation for the model have the following features:

1. We modeled each muscles, tendons, and ligaments as a single wire. By setting an origin, via-points, and an end for each wires, we enabled precise modeling for each components.
2. We can compute forward/inverse dynamics computation of the model.
3. By applying effiecient algoritm for closed kinematic chains, we realize $O(n)$ computational cost for the model which is comprised of n elements.

This forward/inverse dynamics computation of Musculo-Skeletal human model allows us to deal with somatosensory information such as musculortendon tensions, ligament stress, and bone stress, and there are many applications to use them in sports science and medical field as mentioned above. Also, in application to computer graphics, somatosensory information enable us build up precise and natural human motion. These information can also be used for VR applications.

In section 2 we describe the methods to design Musculo-Skeletal human model using closed chain of links and wires, in order to apply the efficient dynamics computation. Then in section 3, we describe dynamics computation method to compute somatosensory information of Musculo-Skeletal human model based on efficient algorithm for closed kinematic chains. Experimental results for inverse/forward computation are described in section 4, followed by the conclusions.

2. Musculoskeletal Model

2.1. Introduction

The detailed human model we designed(Fig.1) is comprised of the skeleton and the musculotendon network. The skeleton is a set of bones grouped into suitable number of body parts. And the musculotendon network is a set of muscles, tendons, and ligaments spreading and straining among bones. Each attribute of these elements is modeled in the following.

Bone Rigid link element with inertia.

Muscle Active constrictive wire actuator.

Tendon Passive constrictive wire that couples with a muscle and transmits its power.

Ligament Passive constrictive wire that connects bones and restrains its relative movement.

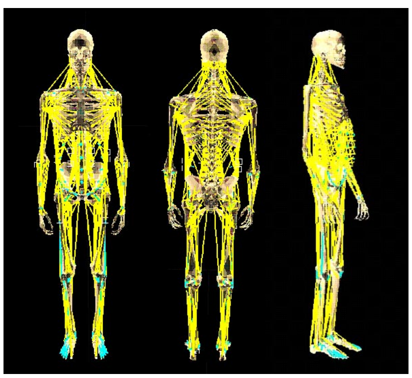

Figure 1: Detailed Human Model

From the next section, we explain the modeling method of the skeleton and the musculotendon network.

2.2. Skeleton Model

We used polygon geometric data of bones on the market for medical science[5]. We also produced polygon data by reconstruction of CT cross-sections. Both are the standard model of European male body.

In the beginning, for an experimentation of the whole body dynamics, we grouped bones under some simplification. In the concrete, the head, hands and foots are counted as one link of the body. Thus facial muscles and short musculotendons on hands and foots are cut out. We do these simplification and grouped about 200 bones into 53 links and modeled all joints as spherical joint.

2.3. Musculotendon Model

There are various parts of muscles, tendons and ligaments. For instance, there are large muscles like triangular muscle, long muscles winding bones and some muscles with furcation like biceps. And they can be classfied into the following design patterns.

1. A part is replaced with a simplest wire which has an origin and an end
2. A part is replaced with a wire which has an origin, via-points and an end
3. A part is replaced with several simple wires
4. A part is replaced with several wires forming a fork at a virtual bone
5. And complex models of those.

We describe actual examples successively

2.3.1. Simple wire model

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

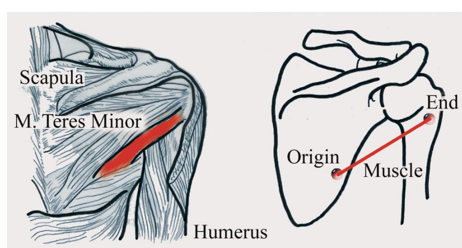

Figure 2: the Model of A Simplest Muscle

2.3.2. Model with via-points

Generally, a wire has an origin, an end and zero or more via-points. We placed via-points at some points where a wire wreathes around a bone or some places where tendons sheaved by a synovial sheath for the purpose of representing rightly points of action and lines of action.

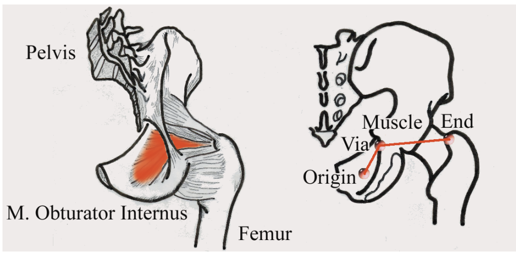

Figure 3: a Muscle with Via-points

2.3.3. Model with virtual bones

Origins, ends or via-points are placed on bones in principle. But, in some cases muscle/tendon/ligament has furcations, we took a virtual link (after-mentioned) and connected wires to it. Fig.5 is the model of bicepsbrachii which has

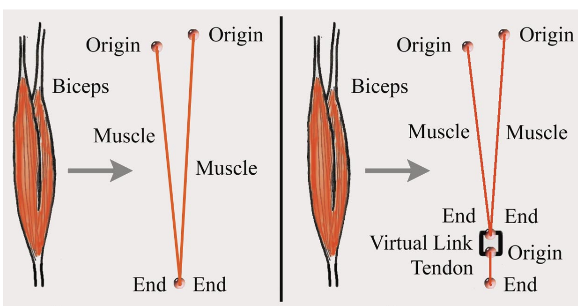

Figure 4: Two Models of Furcation

a bifurcation (and a via-point). It is an example of muscles with furcation. At the last part of bicepsbrachii, the tendon is separating off and each branch of the tendon is forking and inserting to respective bones. Therefore, the furcation model of the tendon is needed to render the function of bicepsbrachii. The reason that virtual links are specially needed to model furcations is ascribable to the fundamental rule that a wire originates from a point and pass through zero or more via-points and ends at a point. This fundamental rule is originally caused by the calculation method of the Jacobian which relates the joint velocities to the differential of lengths of wires. A virtual link doesn't have any mass differently from a bone link. But, as regarding that relative distances among any points on a virtual link are invariable and that a virtual link translate tensions of wires, it can ordinarily be calculated as a link. We describe about these calculation in the next chapter.

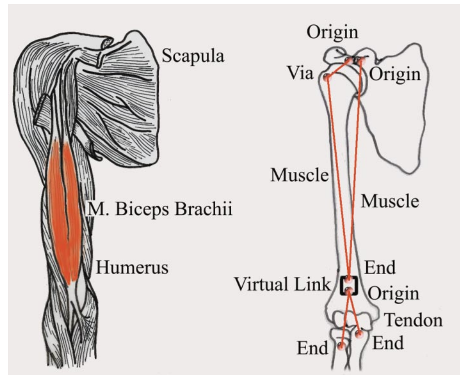

Figure 5: a Model with Bicepsbrachii

2.3.4. Model comprised of several wires

We modeled some large muscles with a number of muscles (e.g. M. Pectoralis Major, Fig.6).

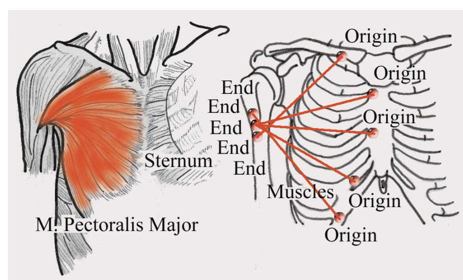

Figure 6: the Model of M. Pectoralis Major

2.3.5. Complex model

Some complex parts can be modeled of combinations of the above patterns. Fig.7 is the example of the ligaments on elbow. Anular Ligament of Radius forms a ring and allows independent pronation/supination of radius and constrains radius and ulna to interlock at flexion/extension of elbow. We take a virtual link and via-points to model the function of it.

There are 547 wires that we designed. We inserted furcations and via-points suitably like this,

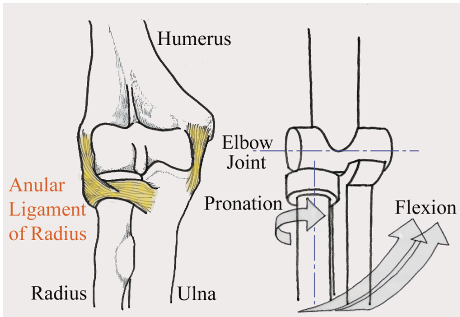

Figure 7: the Mechanism of Elbow Joint

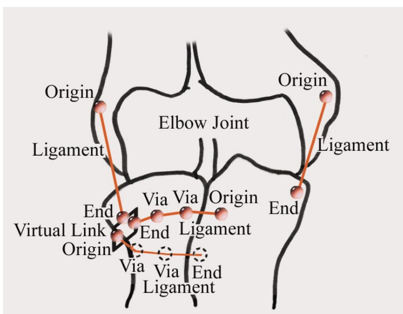

Figure 8: the Model of Ligaments on Elbow Joint

consulting an anatomy book[6].

3. Efficient Dynamics Computation of Musculo-Skeletal Human Model

3.1. Forward Dynamics

When wire tensions(tensile strength for muscles, tendons, and ligaments) are given for a Musculo-Skeletal human model, we can simulate the motion based on the forward dynamics computation. The

Bone	200
Link(group of bones)	53
Virtual Link	28
DOF	153
DOF(includes virtual link)	321
Muscle	366
Tendon	91
Ligament	34
Cartilage	56

Table 1: List of the Number of Elements

computation is based on the following procedures:

1. Compute $\boldsymbol{J} \in \boldsymbol{R}^{N_l \times N_G}$, the Jacobian matrix of wire lengths with respect to the generalized coordinates, as

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

where $\boldsymbol{l}$ is the lengths of wires, $\boldsymbol{\theta_G}$ is the generalized coordinate, and N_l and N_G are the number of wires and the dgree of freedom (DOF) of the chains respectively.

2. Using the Jacobian matrix obtained above, transfer wire tensions into generalized force: in this case joint torques.

3. Compute the motion which will be generated from the joint torques using forward dynamics computation of kinematic chains.

In the following sections, we will describe more precisely about the procedures.

3.1.1. Computation of the Jacobian Matrix

In order to compute the Jacobian matrix of Eq.(1), we first consider $\boldsymbol{J}_{Li}$, the Jacobian matrix of the length of the i-th wire with respect to the generalized coordinate. This matrix forms 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.

Now we describe the computation of $\boldsymbol{J}_{Li}$. We suppose that the i-th element is comprised of $m_i - 2$ ($m_i \geq 2$) intermediate points plus the initial and final point, and let the length between the j-th and $(j+1)$-th via-point $l_{i,j}$, and the Jacobian matrix of $l_{i,j}$ with respect to generalized coordinate $\boldsymbol{J}_{Li,j}$. Then $\boldsymbol{J}_{Li}$ can be described as 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 via-point j, l_i, j can be described in the following form,

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

so differentiating Eq.(4) with generalized coordinate yields the following equation:

$$\boldsymbol{J}_{Li,j} = \frac{\partial l_{i,j}}{\partial \boldsymbol{\theta}_G} \quad (5)$$

$$= \frac{1}{l_{i,j}} \left(\boldsymbol{p}_{i,j+1} - \boldsymbol{p}_{i,j}\right)^T \frac{\partial}{\partial \boldsymbol{\theta}_G} \left(\boldsymbol{p}_{i,j+1} - \boldsymbol{p}_{i,j}\right) \quad (6)$$

$$= \frac{1}{l_{i,j}} \left(\boldsymbol{p}_{i,j+1} - \boldsymbol{p}_{i,j}\right)^T \left(\boldsymbol{J}_{p_{i,j+1}} - \boldsymbol{J}_{p_{i,j}}\right) \quad (7)$$

where $\boldsymbol{J}_{p_{i,j}} = \partial \boldsymbol{p}_{i,j} / \partial \boldsymbol{\theta}_G$, the Jacobian matrix of $\boldsymbol{p}_{i,j}$ with respect to generalized coordinate, and this can be obtained by the computation described in the reference[9]. From Eq.(3) and Eq.(7) we can compute $\boldsymbol{J}_{Li}$. As a result, we can obtain the Jacobian matrix of Eq.(1) by arranging $\boldsymbol{J}_{Li}$ for every wire in a row.

3.1.2. Transferring Wire Tensions into Joint Torques

We compute the joint torques $\boldsymbol{\tau}_G \in \boldsymbol{R}^{N_G}$ from the given wire tensions $\boldsymbol{f}$ which muscles, tendons, and ligaments have produced. $\boldsymbol{f}$ can be determined by the following equation:

$$\boldsymbol{\tau}_G = \boldsymbol{J}^T \boldsymbol{f}. \quad (8)$$

3.1.3. Forward Dynamics Computation of Kinematic Chains

We used the forward dynamics method of kinematic chains which we have proposed in the reference[8]. The method has the following features:

1. For the kinematic model consists of N links, the complexity of the method is $O(N)$, and the complexity can be reduced even more to $O(\log N)$ by applying parallel computation.

2. The same program can be used for parallel computation on any number of processors as well as serial computation.

3.2. Inverse Dynamics computation using Mathematical Programming Opimization

Inverse dynamics computation of detailed human model is to compute the tensile strength of muscles, tendons, and ligaments from the motion data which we can obtain by human motion captures or by motion choreographies. The computation requires the following procedures:

1. Applying inverse dynamics computation of kinematic chains to the given motion data, and obtain joint torques $\boldsymbol{\tau}_G$.

2. Mapping of the joint torques to the wire tensions $\boldsymbol{f}$.

The mapping problem is difficult because the number of wires are quite larger than that of link joints. We used mathematical programming method to solve this redundancy problem. We will describe more precisely in the following.

3.2.1. Basic Idea

We transfer joint torques into tensile strength of muscles, tendons, and ligaments. The relationship between joint torques and wire tensions is represented in Eq.(8). As the number of wires are quite larger than that of link joints, there is a redundanncy problem when we determine wire tensions. The wire tensions only works for contracting direction and this constraint conditions are expressed as

$$\boldsymbol{f} \leq \boldsymbol{0}. \quad (9)$$

The straightforward approach for the mapping is to find the least square solution of $\boldsymbol{f}$ that minimize the cost function

$$Z = \frac{1}{2} \mid \boldsymbol{\tau}_G - \boldsymbol{J}^T \boldsymbol{f} \mid ^2 \quad (10)$$

under Eq.(8) and Eq.(9). However, as there are large number of variables, it is time consuming to solve this least square problem. So we will propose two efficient methods for the mapping computation using mathematical programming methods. First we descirbe the method using linear programming, and then we describe the method using quadratic programming.

3.2.2. The Idea of Formulation for Linear Programming

As the constraint conditions Eq.(8) and Eq.(9) are both linear equations, we will apply the method of linear programming(LP) to the mapping problem. Linear programming is an optimization technique to find an extreme (i.e., minimum or maximum) value of linear objective function, subject to linear equality and inequality constraints. For LP problems, the Simplex method provides fast solutions to very large scale applications, sometimes including hundreds of thousands of variables. We will formulate the mapping problem into a LP problem, and make the computation more efficient.

We define vector $\boldsymbol{a} \in \boldsymbol{R}^{N_l}$ whose elements are all non-negative. Each elements of this vector are costs for corresponding elements of the wire tensions $\boldsymbol{f}$. Now we consider the following linear objective function

$$\boldsymbol{a}^T \boldsymbol{f}. \tag{11}$$

Since all the elements of $\boldsymbol{f}$ are non-positive, maximizing Eq.(11) will minimize the total sum of wire tensions. Finally the mapping of the joint torques to the wire tensions can be replaced to the LP problem which maximize Eq.(11) under constraints Eq.(8) and Eq.(9). The solution will minimize the total sum of wire tensions.

In fact, when we tried to solve the above LP problem, the result was that there existed no solution which satisfies the condition Eq.(8) and Eq.(9). This was due to the joint torques of virtual links and neck vertebras. These joint torques were not fully represented by the wire tensions, because of the modeling limitations. So we defined a vector $\boldsymbol{\delta} \in \boldsymbol{R}^{N_G}$ and modified the equality constraint conditions Eq.(8) into the following inequality conditions:

$$\boldsymbol{\tau}_G - \boldsymbol{\delta} \ \leq \ \boldsymbol{J}^T \boldsymbol{f} \ \leq \boldsymbol{\tau}_G + \boldsymbol{\delta}. \tag{12}$$

$\boldsymbol{\delta}$ represents the permissible range of error for the joint torques, and we chose proper value as small as possible. Then the modified LP problem would be to maximize Eq.(11) under constraints Eq.(9) and Eq.(12).

3.2.3. The Idea of Formulation for Quadratic Programming

In LP method, we permitted some error $\boldsymbol{\delta}$ when we compute wire tensions $\boldsymbol{f}$. In the method using quadratic programming, we will try to find wire tensions $\boldsymbol{f}$ which satisfies Eq.(8) as close as possible. QP is an optimization technique to find an extreme value of the objective function which have the following form:

$$Z = \frac{1}{2}\boldsymbol{x}^T \boldsymbol{Q} \boldsymbol{x} + \boldsymbol{c}^T \boldsymbol{x} \tag{13}$$

under linear equality and inequality constraints. If the matrix $\boldsymbol{Q}$ is a symmetric positive semidefinite matrix, the problem is called convex quadratic programming problem and there are many efficient algorithm for such problem. Now we consider the following quadratic objective function:

$$\begin{aligned} Z &= \mid \boldsymbol{\tau}_g - \boldsymbol{J}^T \boldsymbol{f} \mid ^2 \\ &= \boldsymbol{f}^T \boldsymbol{J} \boldsymbol{J}^T \boldsymbol{f} - 2\left(\boldsymbol{J}\boldsymbol{\tau}\right)^T \boldsymbol{f} + \boldsymbol{\tau}\boldsymbol{\tau}^T. \end{aligned} \tag{14}$$

By setting Eq.(14) as an objective function, we can minimize the error between joint torques and the force from wires.

4. Experimental Results

4.1. Forward Dynamics

In the experiment, we modeled muscles as soft spring and dumper, and tendons and ligaments as hard spring and dumper. We set active muscle force as zero, and simulated motion when we hang the head. Fig.9 is the motion which we obtained in the simulation. The body hanged down according to the gravitational pull. The simulation was

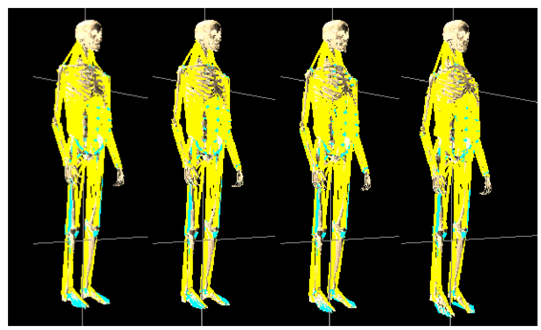

Figure 9: an Example of Forward Dynamics

taken on Pentium IV 2 Ghz machine, and the computational time was $0.5[sec/frame]$. The reason why we went no further than doing experiment of such simple motion is that we have no way of designing the whole body muscle force yet. It is our future work to design controllers for the muscles to generate appropriate motions.

4.2. Inverse Dynamics

In the experiment, we used motion captured data of “kicking” for the motion data and computed wire tensions from the data. For the computation of LP problem, we used the Simplex method routines of GNU free LP library GLPK[10]. For the computation of convex QP problem, we used the Primal-Dual Interior-Point Algorimthms routines of free object-oriented software package for solving convex QP problem(OOQP[11]).

The computational time are shown in the table bellow. The average computational time per frame and the value of Eq.(10), which represents the error for the joint torques, are shown in Table 2.

Table 2: computational time and accuracy

	LP method	QP method
time [sec/frame]	0.375	27.25
Z [N^2]	8,622	2,339

The results for the computation using LP method are shown in comparison to the results for the method using convex QP. All the measurements were taken on Pentium IV, 2 Ghz machine. It is obvious that the method using LP is more efficient than the method using QP. However, in QP method we can minimize the error. We visualized the wire tensions by classifying the force by color. Fig.10 shows the result from the computation.

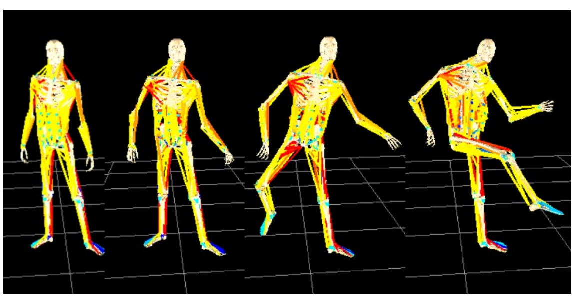

Figure 10: an Example of Inverse Dynamics

In this inverse dynamics methods, we have determined the muscular force by utilizing mathematical programming methods, and we are not sure that the computed force are consistent with the actual human muscular force. In order to reflect the human feature to the dynamic computation, we need to use information from biomechanical experiments. For example, in the LP method, we set all the elements of the cost vector $\boldsymbol{a}$ as 1 and all the elements of the error vector $\boldsymbol{\delta}$ as a an equivalent value. In order to reflect the human feature, we need to design appropriate costs for each muscles with the feedback from biomechanical experiments.

5. Conclusion

The contributions of this paper are summarized as follows:

1. We proposed modeling methods for human dynamic system. In this model, musculo-tendons are computable as kinematic chains. And we designed a detailed human model based on the modeling methods.
2. We developed computational algorithms for inverse/forward dynamics of wire-based human model. The computational algorithms were successfully implemented.

Following problems and extensions are to be involved in future works:

1. Building in the mass properties. Currently, we split it into rigid links arbitrary.
2. Redundancy problem of muscles in inverse dynamics. We are searching for a solution of this problem by introduction of interior model of constriction of muscle and using electromyograph with motion capture at measurement.

Acknowledgements

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

References

[1] Kaya, M., Minamitani, H., Hase, K., Yamazaki, N.: Motion Analysis of Optimal Rowing Form by Using Biomechanical Model, Proc. IEEE Engineering in Medicine and Biology, (1995), 715-716.

[2] Taku Komura, Yoshihisa Shinagawa, 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] 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.

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

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

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

[7] Y. Nakamura and K. Yamane: Dynamics Computation of Structure-Varying Kinematic Chains and Its Application to Human Figures, IEEE Trans. on Robotics and Automation, Vol.16, No.2, pp.124-134, 2000.

[8] Katsu Yamane, Yoshihiko Nakamura: O(N) Forward Dynamics Computation of Open Kinematic Chains Based on the Principle of Virtual Work, Proceedings of IEEE-ICRA Int. Conf., pp2824-2831, 2001.

[9] 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.

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

[11] OOQP: http://www.cs.wisc.edu/~swright/ooqp/