2010 IEEE International Conference on Robotics and Automation
Anchorage Convention District
May 3-8, 2010, Anchorage, Alaska, USA

Joint Proprioception Acquisition Strategy Based on Joints-Muscles Topological Maps for Musculoskeletal Humanoids

Yuto Nakanishi, Kazuo Hongo, Ikuo Mizuuchi and Masayuki Inaba

Abstract— **Many humanoids have been developed, but more complicated and flexible humanoids must be developed, in order to realize more natural and various motions like humans. However, it is difficult to measure directly joint posture in the multi-dofs joint of such robots (e.g. a hip spherical joint) because of its complicated structure. This paper describes an estimation method for tendon-driven joint postures of these complicated multi-dofs joints, only based on information of muscles' length relative displacement data during joints movement. We regard this posture estimation problem as a pattern matching problem in the mapping space from joint posture to muscle lengths and solve this problem by using very simple searching algorithms. Furthermore, this paper describes a strategy for deciding muscles motor command to acquire its joint proprioception by the proposed joint posture estimation method only based on information of muscles' length relative displacement data. Finally, we confirmed the feasibility of the proposed estimation method by applying this algorithm to the real tendon-driven musculoskeletal humanoid Kojiro.**

I. INTRODUCTION

Many humanoids have been developed, but more complicated and flexible humanoids must be developed, in order to realize more natural and various motions like humans. Based on these standpoints, more anthropomimetic musculoskeletal humanoids, which have flexible spine structure and redundant muscles, are studied in these years[1], [2]. These humanlike humanoids have not only simple rotational joints and also complicated joint structures, such as spherical joints like hip joints, oval spherical joints like wrist joints and shoulder compound joints, which consist of a collar bone and a scapulaFig.1. These complicated joints have advantages that they can build up the joint which has more than 2 degree of freedoms by one simple and compact structure, and that they are so strong thanks to its large contact surface. But these joints are difficult to measure their joint posture directly. This disadvantage can be a big problem in robots' motion control which needs self joints' posture. In the case of a rotational joint, its accurate joint posture(i.e. its joint angle) can be easily acquired by calibrating base joint angle(i.e. zero position) when setting the joint to the mechanical joint limitation and measuring its relative joint angle through an accurate rotary encoder. In the case of a spherical joint, its joint relative posture can be estimated by processing spherical joint's surface images from a small cellular phone camera embedded in the spherical joint's socket[3]. In the case of more complicated joints, such as an oval spherical joint(a human wrist joint), however, it is difficult to embed devices which measure joint posture into joint's structure. It is possible to estimate a joint posture by measuring bone frame posture both sides of the joint, using gyroscopes and accelerometers or Hall effect sensors embedded in the bone frames[4]. These methods are not realistic ways, because these measurement sensor devices are quite big and they can often drift. Another problem of these joints posture measurement is that it is difficult to calibrate base joint position due to no explicit joint limitation as in the case of rotational joints.

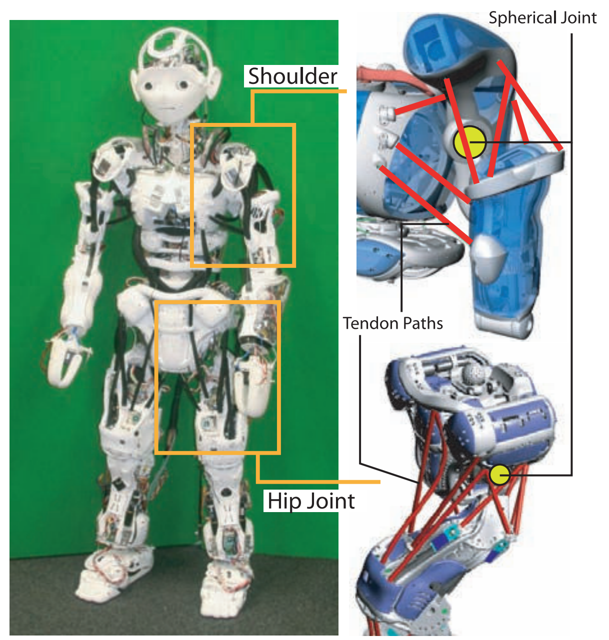

Fig. 1. Spherical joint structures of musculoskeletal humanoid Kojiro

By the way, we human beings have no sensors to measure self joints posture directly, but we can acquire self joint posture sense by integrating body motion information, such as muscle tension and muscle displacement, from organ of proprioception that is goldi tendon organ and neurotendinous spindle. In this paper, this type of joint posture sensation is named "joint proprioception". If musculoskeletal humanoids can acquire joint proprioception from their muscles' motion information like human beings, they need no joint sensor devices and also no base joint pose calibration. it is very useful in developing humanoids which have more humanlike complicated joint structure.

In this paper, we propose how to acquire joint propriocep-

Y. Nakanishi, K. Hongo and M. Inaba are with Department of Mechano-Informatics, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, 113-8656 Tokyo, JAPAN `nakanish,hongo,inaba@jsk.t.u-tokyo.ac.jp`

I. Mizuuchi is with Division of Mechanical Systems Engineering, Tokyo University of Agriculture and Technology, 2-24-16 Nakacho, Koganei, Tokyo, 184-8588, JAPAN `ikuo@mizuuchi.lab.tuat.ac.jp`

tion in humanlike musculoskeletal humanoids' complicated joints. **II** describes a basic idea of a joint proprioception, which is a joint posture estimation method using muscle relative displacement information during robot's joint motion, based on non-linear relationship between joint and muscle space. **III** describes an algorithm for joint posture estimation. In **IV**, we propose a strategy of muscle motion commands generation to acquire joint proprioception using the proposed joint posture estimation method. Finally in **V**, we confirmed that joint proprioception could be acquired using the proposed method, by some experiments of a musculoskeletal humanoid robot Kojiro we developed.

II. MUSCULOSKELETAL JOINT PROPRIOCEPTION

A. Tendon-driven joint structure dealt in this paper

In this paper, we deal tendon-driven joints which have more than 2 DOFs, such as spherical joints and shoulder compound joints, and are driven by redundant muscles whose **relative displacement** can be measured. For example, tendon-driven robots often adopt a wire-pulley driven by a motor actuator as muscle-tendon mechanism. In these robots, each muscle(tendon) length displacement can be indirectly calculated by pulley radius and pulley revolution measured by motor's rotary encoder. It is important to need **no absolute muscle length information**, which is very difficult to measure, for joint posture estimation proposed in this paper.

And also it is desirable to measure each muscle tension information in order to prevent each muscle from coming loose during joint motion. It is because that accuracy of joint proprioception depends on accuracy of muscles relative length measurement.

B. Joint pose estimation by muscles relative displacements

We assume how to estimate posture of the multi-DOFs joint which is driven by m muscles. It is an objective to estimate current joint posture $\boldsymbol{\theta}(t_{now})$ using muscles relative displacements data $\boldsymbol{L}_r = \{l_{r_i}(t)|i = 1..m, t = t_0, ..t_{now}\}$ obtained during this joint motion. $l_{r_i}(t)$ is i muscle relative displacement at time t. Joint posture estimation can be regarded as the problem to estimate position of the trajectory of muscle relative displacement sample data $\boldsymbol{L}_r$ in the topological map between joint posture $\boldsymbol{\theta}$ space and muscle absolute length $\boldsymbol{L}_a$ space, as shown in Fig.2.

Therefore, this joint posture estimation is **one of localization problems**. As localization application, global position estimation for mobile robots has been studied[5]. Basic localization approach is following:

1) Generation of map
2) Getting inner sensor information of a mobile robot (Wheel rotary encoder, laser range finder for measuring distance to walls and obstacles)
3) Identification of global position of a mobile robot based on sensor information and map knowledge

In localization step 3, it is important using landmarks on the map(i.e. a corner of a hall, a post, a hedge and so on) in order to reduce integration error based on wheel rotary encoders[6], [7].

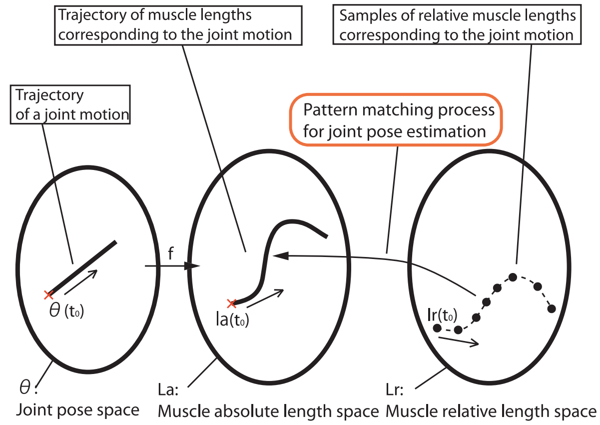

Fig. 2. Image of a joint poseture estimation problem

These localization approach can be applied to joint pose estimation problem in musculoskeletal humanoids as follows:

1) Generation of joints-muscle lengths topological map
2) Getting inner sensor information (muscles relative lengths)
3) Identification of joint pose of a musculoskeletal humanoid

This joint-muscles topological map $f : \boldsymbol{\theta} \rightarrow \boldsymbol{L}_a$ can be easily generated by a geometric musculoskeletal robot model which describes position of joints, size of bones and position of all muscles attachment points. And also the map can be obtained by sensor information of real robot body during actual joint movement. In this case, we have to put joint posture sensor devices, such as gyroscopes and accelerometers, to the robot body temporarily.

C. Joint estimation condition: non-linearity of joint-muscles

In order to estimate joint posture based on only relative muscle displacement information, it is necessary that Jacobian $\boldsymbol{J}_l$ between joint posture and muscle lengths is not constant. In case of the rotational joint mechanism driven by two tendons whose paths are constrained by the pulley, Joint-muscles Jacobian $\boldsymbol{J}_l$ is constant. In this case, we cannot identified where the trajectory of relative muscles displacement sample data $\boldsymbol{L}_r$ is in the map $f : \boldsymbol{\theta} \rightarrow \boldsymbol{L}_a$.

On the other hand, in the case of complicated tendon-driven joint which have complicated tendon paths as shown in Fig.1, Joint-muscles Jacobian $\boldsymbol{J}_l$ is not constant obviously, that is, the joint-muscles relationship has non-linearity. In these joints, we can identify position of the trajectory of $\boldsymbol{L}_r$ in the map. **This non-linearity in joint-muscles topological map is corresponding to landmarks of the map in case of global position estimation problems for mobile robots.**

III. ALGORITHM OF JOINT POSTURE ESTIMATOIN BASED ON MUSCLES RELATIVE LENGTHS DATA

This section describes the algorithm of joint posture estimation based on muscle relative displacements data during joint movement. This algorithm consists of two parts: 1)Joint posture estimation based on current muscle absolute lengths,

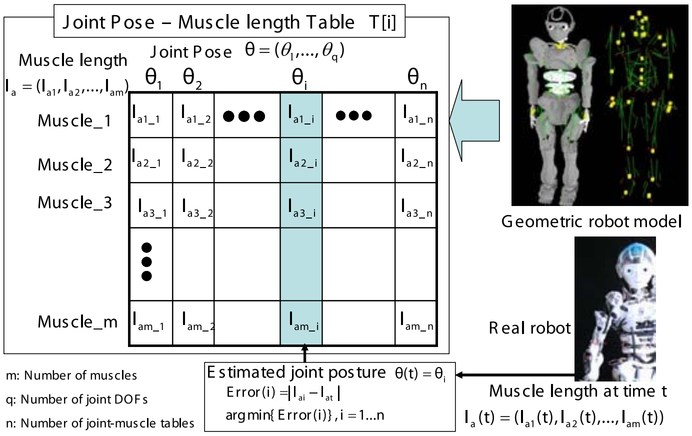

Fig. 3. Joint pose estimation by muscle absolute lengths

2)Base joint posture estimation based on muscle relative displacements time-series data.

A. Joint pose estimation by muscles absolute lengths

Fig.3 shows joint pose estimation flowchart based on muscle absolute lengths. The absolute muscles lengths vector $\boldsymbol{l}_a = (l_{a_1}..l_{a_m})$ at one joint posture vector $\boldsymbol{\theta} = (\theta_1...\theta_q)$ can be easily calculated by using the geometric musculoskeletal robot model which describes positions of joints and muscle point of origin and end, where q denotes joint's DOFs and m denotes the number of muscles. The table $T[i] = (list \quad \boldsymbol{\theta}_i \quad \boldsymbol{l}_{a_i}), \quad i = 1..n$ between the joint posture vector and the muscle absolute length vector can be obtained by quantization of joint posture space and calculation of the each muscle absolute length vector $\boldsymbol{l}_{a_i}$ correspond to each quantized joint posture vector $\boldsymbol{\theta}_i$, where n denotes the number of table elements, that is the number of quantization.

The muscle absolute data vector at time t is defined as $\boldsymbol{l}_a(t)$. If we can find the muscle absolute vector $\boldsymbol{l}_{a_j}$, which has the least vector distance with $\boldsymbol{l}_a(t)$, among generated joint-muscle length table $T[i]$ elements, the joint posture vector $\boldsymbol{\theta}_j$ correspond to $\boldsymbol{l}_{a_j}$ can be regarded as the joint posture $\boldsymbol{\theta}_t$ at time t.

Therefore, joint posture estimation based on muscles absolute lengths can be dealt as a nearest neighbor search problem as follows:

$$Error(i) = |\boldsymbol{l}_{a_i} - \boldsymbol{l}_a(t)| \tag{1}$$

$$argmin\{Error(i)\}, \quad i = 1..n \tag{2}$$

Here, the more joint DOFs is, the more number of quantized table data are generated and the higher computation cost is needed. In Kojiro's shoulder 3 DOFs spherical joint, if the joint posture space is quantized every 1[degree], about 800000 table elements are generated, where dimension of muscle length vector is 5. In this case, it takes about 3[msec] to search nearest neighbor vector among all elements on the Intel Core2Duo 3G[Hz] PC. In this search problem for joint posture estimation, however, vector dataset of the table is fixed as long as muscle attachment position does not be changed. The search time can be drastically reduced by rebuilding the dataset structure which can be effectively searched, such as kdtree structure. Actually it takes 4[μsec] on the same PC condition to do approximate nearest neighbor search(within 200 steps) from kdtree dataset.

B. Base joint posture estimation by muscle relative length time-series data

In the real musculoskeletal humanoids, it is difficult to measure absolute muscle length. In case of Kojiro, whose muscle actuator system adopts wire-pulley mechanism using motor, muscle length is calculated based on pulley radius and pulley revolution by motor's rotary encoder. Therefore, we can obtain only relative muscle displacement $\boldsymbol{l}_r(t)$ at time t. In this time, we do not know absolute muscle length information and we cannot directly use the joint posture estimation method proposed in **III-A**. However, joint posture can be estimated based on the muscle relative length time-series data $\boldsymbol{L}_r = \{\boldsymbol{l}_r(t)|t = t_{now}, ..t_0\}$ during joint movement.

If we assume that the joint posture vector at time $t = t_{now}$ is $\boldsymbol{\theta}_k$(i.e. the k th posture vector of the joint-muscles table $T[i]$), we can convert all relative muscle displacement time-series data $\boldsymbol{L}_r$ into absolute muscle length time-series data $\boldsymbol{L}_{a_k}$ as follows:

$$\boldsymbol{L}_{a_k} = \{\boldsymbol{l}_a(t) = \boldsymbol{l}_r(t) - \boldsymbol{l}_r(t_{now}) + \boldsymbol{l}_{a_k} \\ |t = t_{now}, ..t_0\} \tag{3}$$

In Eq. (3), $\boldsymbol{l}_a(t)$ denotes absolute muscle length vector at time t and $\boldsymbol{l}_{a_k}$ denotes absolute muscle length vector corresponding to the joint posture $\boldsymbol{\theta}_k$ in joint-muscle table $T[i]$ in Fig.3.

Here, we can calculate the summation $E(k)$ of estimation result error $Error(i)$ obtained by joint posture estimation(Eq. (1), Eq. (2)) based on each absolute muscle length vector dataset $\boldsymbol{L}_{a_k}$, as follows:

$$E(k) = \sum_{i=t_0}^{t_{now}} Error(i) \tag{4}$$

If the assumption that the joint posture vector at time $t = t_{now}$ is $\boldsymbol{\theta}_k$(i.e. the k th posture vector of the joint-muscles table $T[i]$) is correct, ideally $E(k)$ will be zero. On the other hand, if the assumption is wrong, $E(k)$ will be greater value. Therefore, $P(i) = 1/E(i)$ can be a likelihood of joint posture estimation. The joint posture at $t = t_{now}$ is the joint posture whose $P(i) = 1/E(i)$ becomes maximum in the joint-muscle topological table $T[i], i = 1..n$. Here, we should take notice that non-linearity relationship between joint-muscles is necessary to estimate joint posture based on relative muscle displacement time-series data as mentioned in **II-C**.

IV. Muscle motion strategy for self acquisition of joint proprioception

In this section, we propose how musculoskeletal robot to generate self muscle motion commands in order to acquire its joint proprioception by itself using the joint posture estimation algorithm proposed in **III-B**. In order to obtain

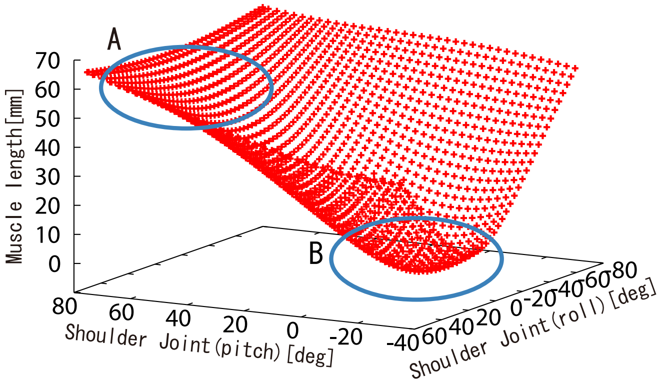

Fig. 4. Relationship between muscle length and joint posture of Kojiro's shoulder

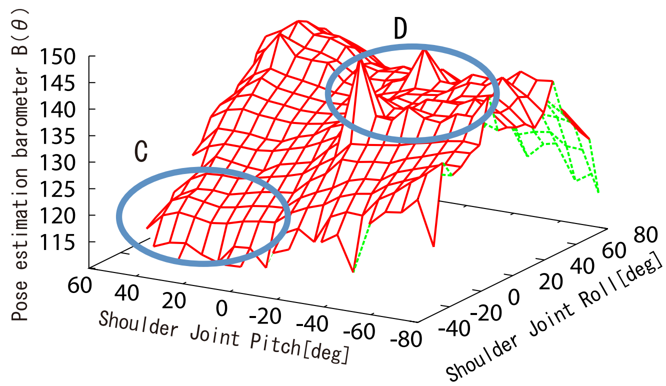

Fig. 5. Map of joint pose estimation barometer in Kojiro shoulder

muscle relative displacements data during joint movement, its joints have to be moved under no joint proprioception where the robot cannot detect its own joint posture. In this case, it only has to give all muscles controlled by tension arbitrary tension target commands. For example, the most simple way is to send a big tension command to only one muscle and small tension commands to the other muscles. By this way, some joint torque can be generated and then the joint should start moving. If the muscle of a big tension command is changed one by one, enough muscle relative length data can be obtained, and joint proprioception can be acquired by applying the joint pose estimation algorithm to these muscle motion information. However, we should notice that accuracy of joint proprioception depends on the joint moving area to obtain these muscle relative length information.

IV-A describes that accuracy of joint proprioception is involved to the nonlinearity distribution of joint-muscle topological map and propose a better muscle motion strategy for self joint proprioception based on it.

A. Relationship between non-linearity distribution in joint-muscle map and accuracy of joint prpprioception

In **II-C**, we mentioned that complicated joints can have non-linear relationship between joint posture and muscle lengths. Fig.4 shows result plotting one muscle's length correspond to joint posture in Kojiro's 3d-spherical shoulder joint. In this graph, xy plane indicates shoulder joint posture space(roll and pitch), and z axis indicates muscle length. Here, yaw axis of joint posture is 0[degree] and yaw axis is cut for viewability. Gradient of this graph indicates Jacobian between joint and muscle length $\boldsymbol{J} = \partial \boldsymbol{l}/\partial \boldsymbol{\theta}$. In this graph, you will notice that the area A is more flat than the area B. Around the area B even if the joint movement is small, the muscle length displacement becomes greater than around the area A. If joint posture is estimated based on the muscle relative displacement data during joint movement around the area A, estimation result will be less robust against measurement noise and less accurate than around the area B. It is very important to select proper joint posture area where muscle length data are collected in order to acquire more accurate and robust joint proprioception.

As the criterion for selection of proper joint posture area, we propose the summation b($\boldsymbol{\theta}$) of square sum of each element of partial differentiation of joint-muscle Jacobian $\boldsymbol{J}$ as follows:

$$V_{ij} = \partial^2 l_j / \partial^2 \theta_i$$
$$b(\boldsymbol{\theta}) = \sum_{i=1}^{m} \sum_{j=1}^{q} (V_{ij} \times V_{ij}) \qquad (5)$$

where m denotes the number of muscles, q denotes DOFs of the joint, l_j means j th muscle's length and θ_i means i th joint posture angle. If muscle relative length data are collected around the area where $b(\boldsymbol{\theta})$ is great, joint posture estimation result will be more robust and accurate.

In order to collect the muscle motion time-series data during joint movement, however, joint moves within some range, such as 10[degree] of each joint posture space. Here, $b(\boldsymbol{\theta})$ is too local criterion around infinite small neighborhood of the joint posture $\boldsymbol{\theta}$. We redefine new criterion $B(\boldsymbol{\theta})$ as following:

$$B(\boldsymbol{\theta}) = \sum_{i} b(\boldsymbol{\theta}_i) \qquad (6)$$

$B(\boldsymbol{\theta})$ is the summation of $b(\boldsymbol{\theta}_i)$ obtained by quantization of $\boldsymbol{\theta}$'s neighborhood for joint movement to collect muscle length data. This $B(\boldsymbol{\theta})$ is corresponding to amount of feature quantity in the joints-muscles topological map. It means that **more accurate joint posture estimation can be done around $\boldsymbol{\theta}$ which has bigger** $B(\boldsymbol{\theta})$. Therefore, we call $B(\boldsymbol{\theta})$ the joint area criterion for joint proprioception acquisition.

B. Verification of the joint area criterion for joint proprioception acquisition using musculoskeletal geometric model

Fig.5 is the graph plotting $B(\boldsymbol{\theta})$ in each joint posture $\boldsymbol{\theta}$ by applying Eq. (6) to geometric simulation model of Kojiro's shoulder spherical joint, where xy plane indicates shoulder joint posture space(roll and pitch) and z axis indicates $B(\boldsymbol{\theta})$) value. In this case, we set quantization granularity 3[degree] and the range of joint movement 10[degree] to calculate $B(\boldsymbol{\theta})$. Around the area C in Fig.5 joint proprioception result will be less robust and accurate, on the other hand, around the area D it will be more robust and accurate.

We prepared two set of muscle relative length time-series data, which are generated when musculoskeletal geometric

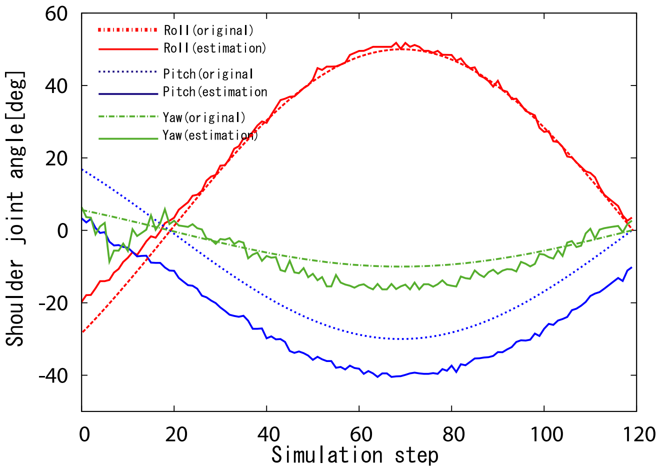

Fig. 6. Bad result of joint pose estimation:around C(Fig.5)

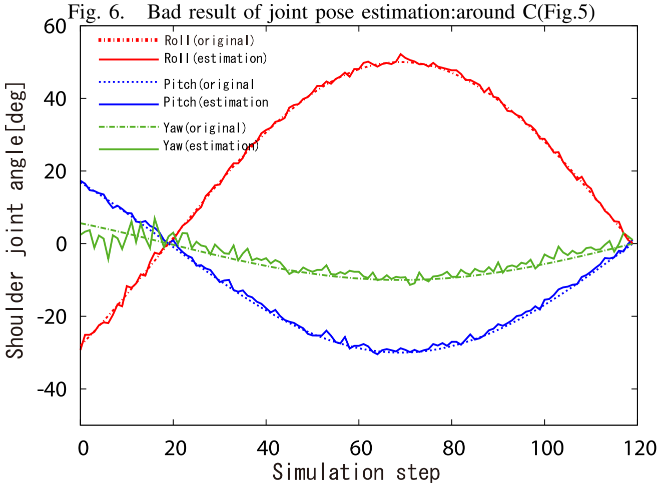

Fig. 7. Better result of joint pose estimation:around D(Fig.5)

simulation model's joint is moved around the area C where shoulder joint posture is roll -20[deg], pitch 20[deg] and yaw 0[deg] and around the area D where shoulder joint posture is roll 20[deg], pitch -40[deg] and yaw 0[deg]. And also white noise(Standard deviation is 1[mm])is added to both of muscle dataset. Next step, we obtained two kind of joint proprioception(typeC and typeD) by applying base joint posture estimation algorithm(**III-B**) to each dataset of muscle lengths.

Finally, we did joint posture estimation. Fig.6 shows the result of joint posture estimation using joint proprioception type C, and Fig.7 shows the result of joint posture estimation using joint proprioception type D. Both joint proprioception were given the same muscle relative length dataset, which were generated during the same joint motion(Signature wave centered roll 0[deg], pitch 0[deg], yaw 0[deg]). In this time, the white noise is added to both datasets. In these graphs, short dashed lines indicate original joint posture trajectories and solid lines indicate estimated joint posture trajectories. Here, we can confirm that the result of joint posture estimation of the joint proprioception(type D) is more accurate than type C one especially in the case of pitch(blue lines) and yaw(green lines). This result shows validity of the joint area criterion(Eq. (6)) for joint proprioception acquisition.

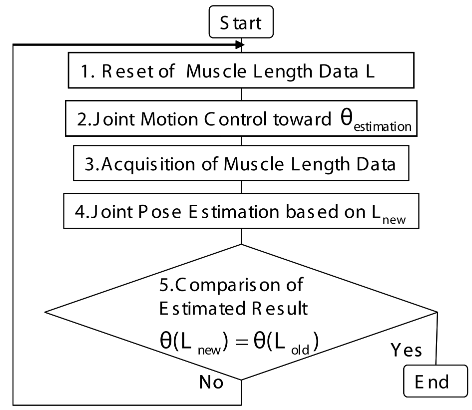

Fig. 8. Joint proprioception acquisition strategy of muscle motion

C. *Muscle motion commands strategy based on the joint area criterion for joint proprioception acquisition*

As mentioned in previous subsections, it is very important the joint area where muscle relative length dataset should be collected in order to acquire joint proprioception. And we can use degree of the non-linearity distribution of joint-muscle topological maps, as the criterion for selection of joint area. According to this findings, we propose the strategy of muscle motion commands generation for joint proprioception acquisition as follows:

1) Reset of muscle relative displacement time-series dataset l_{data}
2) Moving joint under tension control toward $\boldsymbol{\theta}_{estimation}$ according to the joint area criterion $B(\boldsymbol{\theta})$
3) Collecting muscle relative displacement time-series dataset l_{data}
4) Acquisition of joint proprioception based on l_{data}
5) Comparison between estimated joint result of the newest joint proprioception $\boldsymbol{\theta}(\boldsymbol{L}_{new})$ and the result of the former one $\boldsymbol{\theta}(\boldsymbol{L}_{old})$
   If $\boldsymbol{\theta}(\boldsymbol{L}_{new}) = \boldsymbol{\theta}(\boldsymbol{L}_{old})$ then go to end, else go to 1).

In this strategy, self joint movement and acquisition joint proprioception is repeated alternately until the joint moves around the goal joint posture $\boldsymbol{\theta}_{estimation}$. Therefore constant joint proprioception will be acquired without depending on initial joint posture.

V. Experiment of joint proprioception acquisition in Kojiro of the real world

Actually Kojiro's shoulder spherical joint proprioception was acquired by using the proposed strategy of muscle motion command generation. Kojiro's shoulder joint is the 3 DOFs spherical joint which is driven by redundant 5 muscles. Each muscle is a pulley-wire mechanism driven by a motor and only muscle **relative** length displacement can be calculated from its motor's rotary encoder and pulley radius.

At first, we generated the joint-muscle topological map f from geometric musculoskeletal humanoid model in this experiment. The map's joint posture space range is roll(from -30[deg] to 80[deg]), pitch(from -80[deg] to 50[deg]) and yaw(from -25[deg] to 25[deg]). and the map's quantization

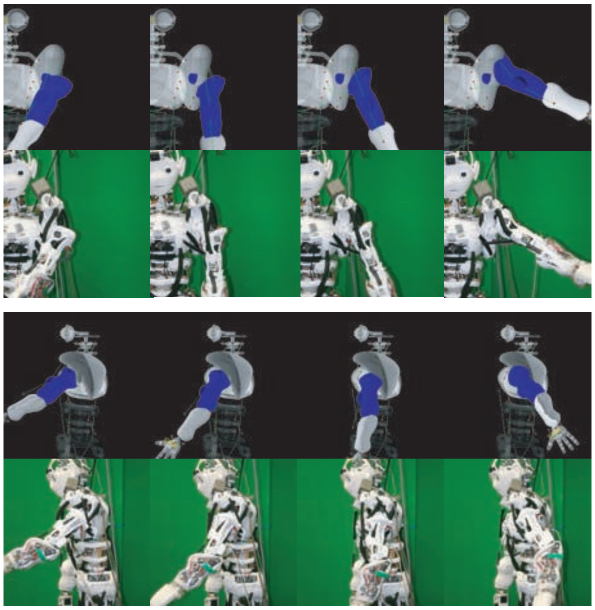

Fig. 9. Comparison of joint postures between the real robot and the estimated result in the case of Kojiro left shoulder sherical joint

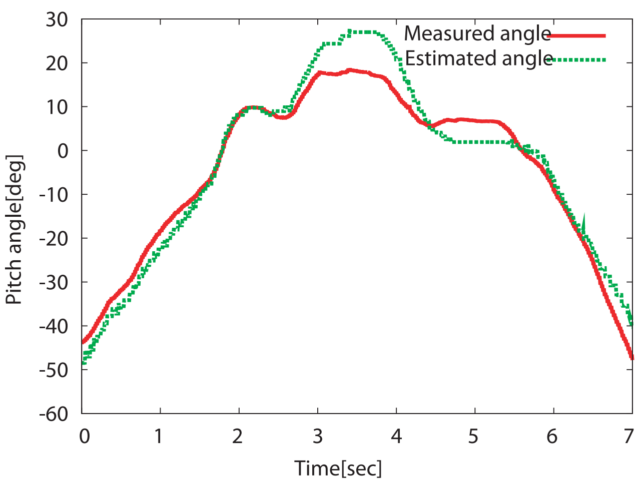

Fig. 10. Comparison of joint postures between the real robot and the estimated result in the case of Kojiro left shoulder sherical joint

granularity is 1[deg]. Next step, according to the joint proprioception acquisition strategy of muscle motion, joint proprioception was acquired. At this time, $\theta_{estimation}$ in Fig.8 is set to 20[deg], pitch -40[deg] and yaw 0[deg], where is the area D in Fig.5.

Fig.9 shows demonstration of joint posture estimation using joint proprioception acquired. We confirmed that the result of joint posture estimation was quite good. At this demonstration, we compared the result of the joint posture trajectory based on the joint proprioception acquired and the trajectory measured by 3-dimentional position and posture sensor device[1] put on the scapula and the upperarm bone. The result is Fig.10. In this graph, we confirmed that the estimation error was within 1[degree] in most joint posture area. It can be said that joint proprioception is acquired enough in real robot body. Around the center of Fig.10, however, the estimation error becomes quite big. It is because that the error between geometric robot model and real robot body occurs around this area, where tendon paths and bone frames collide against each other. This problem will be solved by generating the joint-muscle topological map from the real robot body's sensors with external joint posture measurement devices temporarily.

[1]which is based on magnetic tracking system(POLHEMUS, 3SPACE FASTRAK)

VI. CONCLUSIONS AND FUTURE WORKS

In this paper, we proposed how to acquire joint proprioception in humanlike musculoskeletal humanoids' complicated joints, which are difficult to embed devices to measure joint posture directly. A basic idea of a joint proprioception is a joint posture estimation method using muscle relative displacement information during robot's joint motion. The point of this estimation method is to focus non-linear relationship between joint and muscle space and muscle redundancy, which are also negative complicated characteristics of musculoskeletal humanoids and also can be rich information to estimate the joint posture based on the muscle length displacements.

Furthermore, we proposed a strategy of muscle motion commands generation to acquire joint proprioception using the proposed joint posture estimation method and we implemented the proposed method into the real musculoskeletal humanoid's system and we confirmed the feasibility and usefulness of the proposed method in some experiments around the Kojiro's shoulder. This self acquisition of joint proprioception algorithm is innovative to musculoskeletal humanoids which have very humanlike complicated multi-DOFs joints. Thanks to this algorithm, it becomes unnecessary to do troublesome joint muscle calibration or to put joint posture sensors which are expensive, big and often drift.

REFERENCES

[1] Owen Holland and Rob Knight. The Anthropomimetic Principle. In *Proceedings of the AISB06 Symposium on Biologically Inspired Robotics*, 2006.

[2] Ikuo Mizuuchi, Yuto Nakanishi, Yoshinao Sodeyama, Yuta Namiki, Tamaki Nishino, Naoya Muramatsu, Junichi Urata, Kazuo Hongo, Tomoaki Yoshikai, and Masayuki Inaba. An Advanced Musculoskeletal Humanoid Kojiro. In *Proceedings of International Conference on Humanoid Robots(Humanoids'07)*, 12 2007.

[3] Junichi Urata, Yuto Nakanishi, Akihiko Miyadera, Ikuo Mizuuchi, Tomoaki Yoshikai, and Masayuki Inaba. A three-dimensional angle sensor for a spherical joint using a micro camera. In *Proceedings of The 2006 IEEE International Conference on Robotics and Automation*, May 2006.

[4] Andreas Haberli. 2D magnetic microsensor with on-chip signal processing for contactless angle measurement. *IEEE International Solid-State Conference Digest of Technical Papers*, Vol. 39, pp. 332–333, 1999.

[5] W. BURGARD. Integrating global estimation and position tracking for mobile robots : The dynamic markov localization approach. pp. 730–735, 1998.

[6] S. KOTANI. Mobile robot navigation based on vision and dgps information. *International Conference on Robotics and Automation*, pp. 2524–2529, 1998.

[7] K. T. SUTHERLAND. Localizing in unstructured environments : Dealing with the errors. *IEEE Trans. on Robotics and Automation*, Vol. 10, No. 6, pp. 740–745, 1994.