2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)
Beijing, China, November 6-9, 2018

A Method of Joint Angle Estimation Using Only Relative Changes in Muscle Lengths for Tendon-driven Humanoids with Complex Musculoskeletal Structures

Kento Kawaharazuka, Shogo Makino, Masaya Kawamura, Yuki Asano, Kei Okada and Masayuki Inaba

Abstract— **Tendon-driven musculoskeletal humanoids typically have complex structures similar to those of human beings, such as ball joints and the scapula, in which encoders cannot be installed. Therefore, joint angles cannot be directly obtained and need to be estimated using the changes in muscle lengths. In previous studies, methods using table-search and extended kalman filter have been developed. These methods express the joint-muscle mapping, which is the nonlinear relationship between joint angles and muscle lengths, by using a data table, polynomials, or a neural network. However, due to computational complexity, these methods cannot consider the effects of polyarticular muscles. In this study, considering the limitation of the computational cost, we reduce unnecessary degrees of freedom, divide joints and muscles into several groups, and formulate a joint angle estimation method that takes into account polyarticular muscles. Also, we extend the estimation method to propose a joint angle estimation method using only the relative changes in muscle lengths. By this extension, which does not use absolute muscle lengths, we do not need to execute a difficult calibration of muscle lengths for tendon-driven musculoskeletal humanoids. Finally, we conduct experiments in simulation and actual environments, and verify the effectiveness of this study.**

I. INTRODUCTION

Tendon-driven musculoskeletal humanoids [1], [2], [3], [4], which imitate not only the proportion but also the bone and muscle structures of human beings, have been developed vigorously. They are expected to play an active part in the future because of their contact softness, variable stiffness, flexible spine, etc. We can apply the human reflex or sensor system to these humanoids and make use of the obtained results for human beings.

As shown in Fig.1, tendon-driven musculoskeletal humanoids have ball joints like the shoulder and scapula, and structures with multiple degrees of freedom (multi-DOFs) like the spine and neck, etc., in which encoders cannot be installed. Therefore, their joint angles cannot be directly measured by encoders like in ordinary axis-driven humanoids, and joint angles need to be estimated from the changes in muscle lengths. On the other hand, there are studies to directly estimate angles of ball joints using cameras [5], but we do not consider them in this study. In previous studies, methods using table-search [6], extended kalman filter (EKF) with polynomial expression [7], and the EKF with neural network (NN) [8] have been proposed. In these methods,

The authors are with the Department of Mechano-Informatics, Graduate School of Information Science and Technology, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo, 113-8656, Japan. [kawaharazuka, makino, kawamura, asano, k-okada, inaba]@jsk.t.u-tokyo.ac.jp

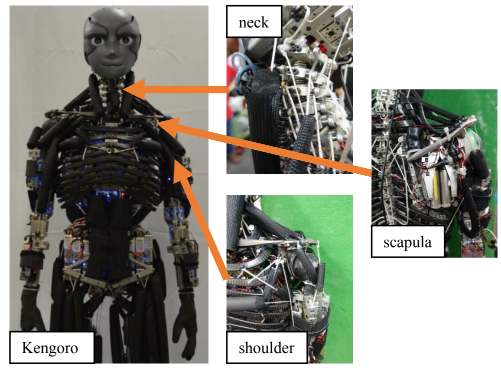

Fig. 1. Complexity of tendon-driven musculoskeletal humanoids. This figure shows that the neck and shoulder girdle of Kengoro have very complex structures like human beings.

we invariably need joint-muscle mapping (JMM), which expresses the nonlinear relationship between joint angles and muscle lengths. Usually, we generate a data set of joint angles and their corresponding muscle lengths, and train JMM with it. However, there is a problem in terms of computational complexity. Tendon-driven musculoskeletal humanoids have numerous polyarticular muscles because they imitate human beings. Therefore, for example, when we express muscle lengths using polynomials of joint angles, the larger the number of polyarticular muscles that cross joints are, the harder the polynomial regression becomes computationally. There are similar problems in the methods using a data table and neural network. In particular, to construct JMM of the whole body is realistically difficult. To solve this problem, we reduced unnecessary DOFs, divided joints and muscles into several groups, and formulated a joint angle estimation method using the JMM that includes polyarticular muscles. By these methods, we solved the problem of computational cost and realized stable joint angle estimation.

Also in this study, we propose a new joint angle estimation method using not absolute muscle lengths but only the relative changes in muscle lengths. In a previous study, Nakanishi, et al. [6] developed a joint angle estimation method using only the relative changes in muscle lengths through table-search. However, this method required a large computational cost and was difficult to apply when the number of DOFs was large. Thus, in this study, we extend the method of Ookubo, et al. [7] and show that the joint angle estimation using only the relative changes in muscle lengths

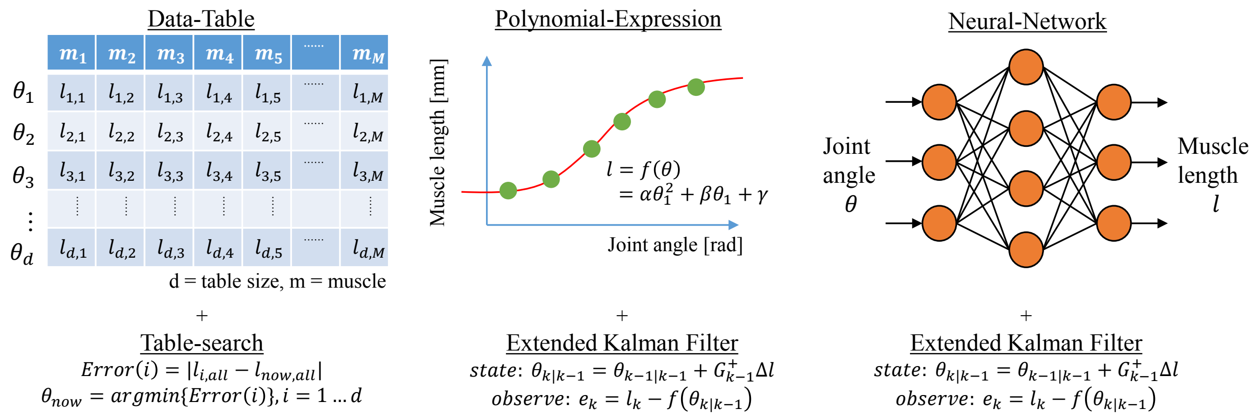

Fig. 2. Overview of the joint angle estimation methods: table-search [6], EKF with polynomial regression [7], and EKF with neural network [8].

is possible. This method makes the calibration of muscle lengths, which is difficult for tendon-driven musculoskeletal humanoids, unnecessary, and the self-body posture can be obtained during exercise. This study can be applied to robots using tendon structures such as the ACT Hand [9], the tensegrity robot [10], and exoskeletons [11]. Also, we hope that this is one step for tendon-driven musculoskeletal humanoids to catch up to ordinary axis-driven humanoids.

This paper is organized as follows. In Section I, we stated the motivation and goal of this study. In Section II, we will summarize the previous joint angle estimation methods and clarify their problems. In Section III, to solve these problems, we will discuss the reduction of unnecessary DOFs, division of joints and muscles into several groups, and formulation of a joint angle estimation method that includes polyarticular muscles. In Section IV, we will extend the previous joint angle estimation method and propose a new joint angle estimation method using only the relative changes in muscle lengths. In Section V, we will conduct experiments using these proposed methods in simulation and actual environments, and verify their effectiveness. Finally, in Section VI, we will state the conclusion and future works.

II. Joint Angle Estimation Methods in Tendon-driven Musculoskeletal Humanoids and Their Problems

First, we will summarize previous studies on joint angle estimation for tendon-driven musculoskeletal humanoids. Next, we will state the problems on the construction of JMM, which is invariably necessary for these estimation methods. Also, we will consider a joint angle estimation method that does not use the absolute muscle lengths but only the relative changes in muscle lengths.

A. Joint Angle Estimation in Tendon-driven Musculoskeletal Humanoids

Because encoders cannot be installed in complex musculoskeletal structures such as ball joints, the scapula, and the spine, we need to estimate joint angles from the changes in muscle lengths. In this study, we do not consider the use of a motion capture system, because it is not available for outdoor use and requires large facilities. We show the summary of previous joint angle estimation methods in Fig.2.

Nakanishi, et al. developed the method using table-search [6]. In this method, JMM is constructed as a data table of the joint angles and muscle lengths of a geometric model, and the current joint angles are obtained from the current muscle lengths by using pattern matching. In this study, a geometric model is expressed with each muscle connecting the start point, relay points, and end point linearly. However, because this method does not use the time series aspect of the data, the continuity of joint angles is not ensured, the precision and computational cost depend on the size of the data table, and it is susceptible to noise.

Next, Ookubo, et al. proposed the joint angle estimation method using extended kalman filter (EKF) with polynomial regression [7]. In this method, JMM is constructed using polynomials of joint angles to express muscle lengths in the geometric model, and the muscle jacobian is obtained by differentiating the JMM. The EKF predicts current joint angles by using the inverse matrix of the muscle jacobian and the relative changes in muscle lengths, and corrects the estimated joint angles by comparing muscle lengths of the actual robot with the estimated muscle lengths, which are calculated from the JMM and predicted joint angles. Also, Kawaharazuka, et al. estimated joint angles by replacing the polynomials with a neural network [8]. The benefit of using a neural network is that the data structure is suitable for the update of JMM. The state and observation equation of the EKF are shown below,

$$\boldsymbol{\theta}_{k|k-1} = \boldsymbol{\theta}_{k-1|k-1} + G^{+}(\boldsymbol{\theta}_{k-1|k-1})\delta \boldsymbol{z}_k \quad (1)$$

$$\boldsymbol{z}_k = f(\boldsymbol{\theta}_{k|k-1}) \quad (2)$$

where $\boldsymbol{\theta}$ is joint angles, G is muscle jacobian, G^{+} is a pseudo inverse matrix of G, $\delta \boldsymbol{z}$ is the relative changes in muscle lengths of the actual robot, $\boldsymbol{z}$ is absolute muscle lengths of the actual robot, and f is JMM.

It is essential that all estimation methods use JMM, stated above, and the construction of JMM greatly affects their computational complexity and precision.

$$\boldsymbol{l} = f(\boldsymbol{\theta}) \tag{3}$$

B. Problems in Construction of the Joint-Muscle Mapping

We showed three joint angle estimation methods using JMM expressed by the data table, polynomials, and NN. In this subsection, we will explain the advantages and disadvantages of these structures.

First, expressions using data tables are superior in terms of simplicity. However, when we use data tables, problems occur in terms of computational cost. When the number of DOFs of the robot is D, and the required precision is of which when the movable ranges of the DOFs are divided into N parts, the size of the data table is N^D. For example, when the number of DOFs of the shoulder is 3, the range of joint angles is 180 deg, and we need a resolution of 0.1 deg, we need a data table the size of 1800^3. Of course, the more the resolution and the number of DOFs increase, the more the size of the data table increases. Therefore, this method is not practical for tendon-driven musculoskeletal humanoids with multi-DOFs like Kengoro [4].

Next, regarding the method using EKF with polynomial regression, because JMM is calculated as polynomials, with a degree of P, a smooth mapping can be obtained, JMM can be differentiated analytically, and the precision is better than in the method using a data table. Also, this method does not need a large memory like with the data table, and we only need the coefficients of the polynomials. However, at the same time, the calculation of the data set construction and polynomial regression require a large computational cost. In the data set construction of this method, we divide the movable ranges of joint angles into N parts, move all DOFs, make all combinations, obtain muscle lengths at each posture, and construct a data set the size of $n = N^D$. Because polynomial regression can compensate for the values among the data set compared to the method using a data table, the N of this method needs to be only about 6–9. When the number of muscles and DOFs included in the JMM is M and D, respectively, we solve the polynomial regression to calculate C as shown below and the computational complexity to calculate it is $O((N^D)^2{}_{D+1}H_P)$ (H means combination with repetition).

$$\begin{pmatrix} l_1^1 & \cdots & l_M^1 \\ \vdots & \ddots & \vdots \\ l_1^n & \cdots & l_M^n \end{pmatrix} = C \begin{pmatrix} \prod_{i=\{0\ldots0\}}^N \theta_i^1 & \cdots & \prod_{i=\{D\ldots D\}}^N \theta_i^1 \\ \vdots & \ddots & \vdots \\ \prod_{i=\{0\ldots0\}}^N \theta_i^n & \cdots & \prod_{i=\{D\ldots D\}}^N \theta_i^n \end{pmatrix} \tag{4}$$

In this equation, l_i^j is the i-th muscle of the j-th data, θ_i^j is the i-th DOF of the j-th data, and θ_0 is 1 for simplicity. For example, when N is 8 and D is 10, the size of the data set is about $1E+9$ (to generate each data takes about 10 msec) and to construct the data set takes a long time. When P is 5, the computational cost to calculate the polynomial regression is about $1E+21$, and it is realistically difficult.

Finally, the computational cost of the method using a NN is almost the same as in the method using polynomials. A bottleneck of the computational cost occurs when constructing the data set and training a NN with it.

From these considerations, we need to group joints and muscles into small parts, and construct JMM regarding each body part. At the same time, because polyarticular muscles cross JMMs, each JMM includes DOFs duplicated in several JMMs (we call a JMM like this polyarticular-JMM), and we need to cope with this problem.

C. Joint Angle Estimation Using Only Relative Changes in Muscle Lengths

Nakanishi, et al. stated that not only the joint angle estimation method using the absolute muscle lengths (absolute-JAE) but also the estimation method using only the relative changes in muscle lengths (relative-JAE) [6] is possible by expressing JMM as a data table. This method uses the nonlinear feature of JMM and calculates the current joint angles by ensuring consistency in time series data of muscle lengths. However, because the method must scan all the spaces of the data table, the larger the number of DOFs is, the larger the computational cost becomes, so this method cannot be used. Therefore, in this study, we extend the joint angle estimation method proposed by Ookubo, et al. [7], and try to estimate joint angles using only the relative changes in muscle lengths. This method can solve problems in computational cost, precision, and the size of data.

In this study, the absolute muscle lengths mean the values of muscle lengths when they are calibrated to 0 at the initial posture (all joint angles are 0 deg). Although these muscle lengths are relative to those of the initial posture in a sense, we call these muscle lengths absolute muscle lengths for simplicity. Also, we assume that each muscle length is measured by an incremental encoder equipped in each muscle motor.

III. Joint Angle Estimation of Complex Musculoskeletal Structures

In this section, we will explain how we should group joints and muscles and construct each polyarticular-JMM, and how we should formulate the joint angle estimation method using these polyarticular-JMMs, when the robot has many complex structures like human beings, such as the scapula and polyarticular muscles. The procedure is shown as below and we will explain each step.

1) reduce unnecessary DOFs of joints to make the size of the JMM small.
2) divide joints and muscles into several groups, and construct polyarticular-JMMs regarding each group.
3) change the formulation of the previous joint angle estimation so as to cope with the DOFs that are duplicated in several polyarticular-JMMs.

A. Reducing DOFs of the Upper Limb

We will state the construction method of the polyarticular-JMM. We treat Kengoro as one example of a tendon-driven musculoskeletal humanoid, and show the relationship between the joints and muscles of Kengoro [4] in Fig.3.

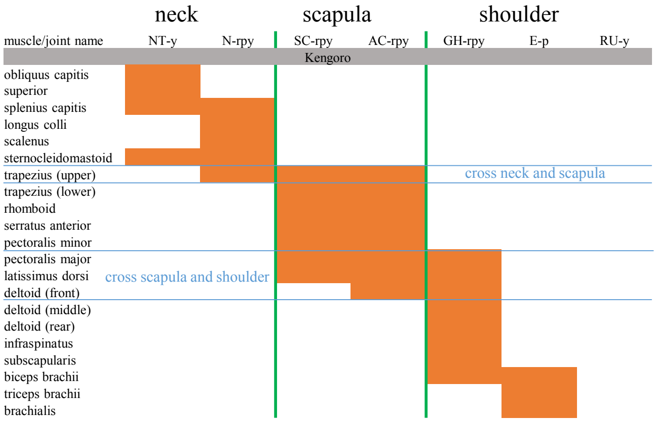

Fig. 3. Correspondence between joints and muscles of Kupper limb of Kengoro. The joints that each muscle can move are painted orange. NT is the neck top 1-DOF joint, N is the neck 3-DOF joint, SC is the sternoclavicular joint, AC is the acromioclavicular joint, GH is the glenohumeral joint, E is the elbow joint, and RU is the radioulnar joint.

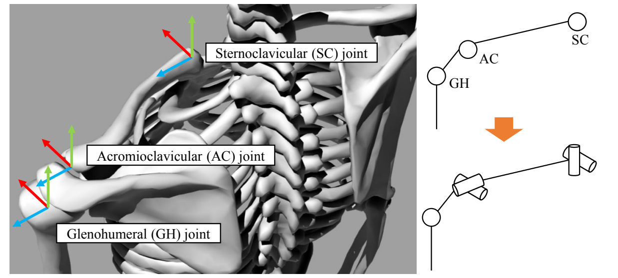

Fig. 4. Details of the scapula structure. The right figure shows the change in DOFs arrangement.

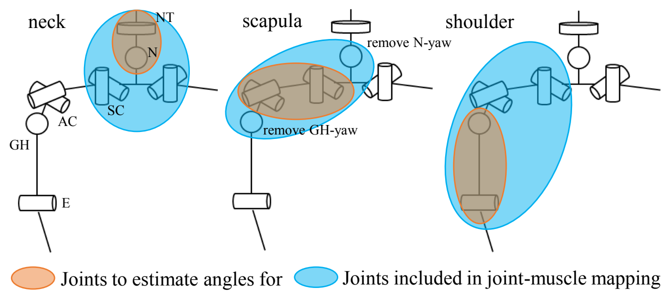

Fig. 5. An example of joint-muscle mapping construction.

First, we choose joints to estimate joint angles for, and obtain the muscles that move the joints. Second, we pick up all the joints that these muscles can move. Finally, we construct JMM using these joints and muscles. For example, when we want to estimate the joint angles of the 3-DOF shoulder, at first, we pick up the muscles that can move the 3-DOF joint. The muscles include polyarticular muscles such as the biceps brachii, pectoralis major, latissimus dorsi, etc., so we pick up all the joints that these muscles can move. The joints include the elbow and scapula joints, and finally, we construct the JMM between 10 muscles and 10-DOF (the 3-DOF shoulder, 1-DOF elbow, and 6-DOF scapula). By using the same procedure, the polyarticular-JMM of the neck joint is the JMM between 10 muscles and 16-DOF.

One of the most important reasons why the number of DOFs included in a polyarticular-JMM become large is that the number of DOFs of the scapula joint(the SC and AC joints), is 6, which is very large. It is clear that the construction of polyarticular-JMMs becomes easier by reducing the number of DOFs of the scapula.

Therefore, we will consider the structure of the scapula first. The detailed structure of the scapula is shown in Fig.4. In the SC joint (roll, pitch, yaw) and AC joint (roll, pitch, yaw), the pitch of the SC joint and the yaw of the AC joint do not move very much. To be precise, we can prevent the scapula from coming off of the thorax by using the pitch of the SC joint or the pitch of the AC joint. Also, the yaw of the AC joint moves only about a maximum of 10 deg as shown in [12]. From these facts, we believe that we can reduce the number of DOFs of the scapula joint. Even though this reduction is not completely accurate, because the motions of the scapula can be expressed roughly, we reduce the number of DOFs of the scapula joint to 4 to make the construction of polyarticular-JMMs easier in this study.

B. Dividing Joints and Muscles into Several Groups

By dividing joints and muscles into several groups and constructing a polyarticular-JMM for each group, the construction of whole body JMM can become easier. Although how the joints and muscles are grouped does not matter, in terms of maintenance, it is better to construct a few number of polyarticular-JMMs in which as many joints and muscles are included as to the limit of computational cost, than to construct a large number of small polyarticular-JMMs. does not matter As one example, we show the grouping of joints and muscles in upper limb of Kengoro in Fig.5. Although we cannot divide joints and muscles simply due to the influence of polyarticular muscles such as the biceps brachii, pectoralis major, trapezius, and latissimus dorsi, we can divide JMM into small polyarticular-JMMs by permitting the duplication of DOFs across several polyarticular-JMMs. Also, in terms of computational cost, we limited the number of DOFs included in each polyarticular-JMM to 8. The result of the grouping is shown in Fig.5. The polyarticular-JMM of the neck includes the roll and yaw DOFs of the right and left scapula AC joints. The trapezius and deltoid (front) cannot move the roll and pitch DOFs of the SC scapula joint very much, so we reduced these unnecessary DOFs as in the theory stated in Subsection III-A. Also, the polyarticular-JMM of the scapula include the roll and pitch DOFs of the neck and shoulder. Like in the neck, we reduce the yaw DOF of the neck and shoulder from the scapula polyarticular-JMM, because muscles around the scapula do not move these DOFs very much. The polyarticular-JMM of the shoulder and elbow includes the roll and yaw of the AC scapula joint and the roll and pitch of the SC scapula joint.

In general, we should set a limitation to the number of DOFs included in each polyarticular-JMM in terms of computational cost, and choose joints and muscles so as not to exceed this limit. In this JMM construction method, polyarticular-JMMs include DOFs duplicated among several polyarticular-JMMs, and we cope with this next.

C. Formulation of a Joint Angle Estimation Method Using Polyarticular Joint-Muscle Mapping

In this subsection, we will discuss the formulation of the joint angle estimation method using polyarticular-JMM by modifying a previous work [7]. In this method, there is a group of joints for which the angles need to be estimated, and a group of joints for which this is not necessary. For example in Kengoro, the polyarticular-JMM used to estimate the 4-DOF scapula joint angles includes two neck DOFs and two shoulder DOFs in addition to the 4-DOF scapula joint. The group of joints we want to estimate angles for, $\boldsymbol{\theta}_y$, is the 4-DOF scapula joint, and the group of joints we do not need to estimate angles for, $\boldsymbol{\theta}_n$, is the 4-DOF of the neck and shoulder. The formulation of the polyarticular-JMM is shown as below.

$$\boldsymbol{l} = f(\boldsymbol{\theta} = \begin{pmatrix} \boldsymbol{\theta}_y \\ \boldsymbol{\theta}_n \end{pmatrix}) \tag{5}$$

The state equation of EKF (Eq.1) in the previous joint angle estimation method [7] is expressed as below.

$$\boldsymbol{\theta}_{k|k-1} = \begin{pmatrix} \boldsymbol{\theta}_y \\ \boldsymbol{\theta}_n \end{pmatrix}_{k|k-1} = \boldsymbol{\theta}_{k-1|k-1} + G^{+}(\boldsymbol{\theta}_{k-1|k-1})\delta\boldsymbol{z}_k \tag{6}$$

In this case, it is easy for $G^{+}(\boldsymbol{\theta}_{k-1|k-1})\delta\boldsymbol{z}_k$ to become unexpected, because $\boldsymbol{\theta}_{n,k|k-1}$ can be predicted using not only the displacement of the polyarticular muscles included in the polyarticular-JMM, but also other monoarticular muscles included in the other JMMs. Therefore, we modify the formulation as shown below, and remove the prediction of $\boldsymbol{\theta}_n$.

$$\boldsymbol{\theta}_{k|k-1} = \begin{pmatrix} \boldsymbol{\theta}_y \\ \boldsymbol{\theta}_n \end{pmatrix}_{k|k-1} = \boldsymbol{\theta}_{k-1|k-1} + \begin{pmatrix} I_y & O_n \end{pmatrix} G^{+}(\boldsymbol{\theta}_{k-1|k-1})\delta\boldsymbol{z}_k \tag{7}$$

Also, because $\boldsymbol{\theta}_n$ is the joint angles obtained by the joint angle estimation using the other polyarticular-JMMs, after the calculation of kalman filter, $\boldsymbol{\theta}_{k|k}$ should be overwritten. If this is not done, $\boldsymbol{\theta}_n$ can become gradually larger or vibrate, and badly influence the estimation of $\boldsymbol{\theta}_y$, because $\boldsymbol{\theta}_{n,k|k}$ is calculated only from polyarticular muscles which cross other DOFs. So, we overwrite $\boldsymbol{\theta}_{n,k|k}$ as shown below.

$$\boldsymbol{\theta}_{n,neck,k|k} = \boldsymbol{\theta}_{y,neck,k|k} \tag{8}$$
$$\boldsymbol{\theta}_{n,scapula,k|k} = \boldsymbol{\theta}_{y,scapula,k|k} \tag{9}$$
$$\boldsymbol{\theta}_{n,shoulder,k|k} = \boldsymbol{\theta}_{y,shoulder,k|k} \tag{10}$$

By these methods, we can estimate joint angles using polyarticular-JMM.

IV. Joint Angle Estimation using Only the Relative Changes in Muscle Lengths

In this section, we extend the joint angle estimation method using the absolute muscle lengths (absolute-JAE) [7], and propose a method that estimates the current joint angles using only the relative changes in muscle lengths (relative-JAE). In the EKF, absolute muscle lengths are needed for the observation equation (Eq.2). In this study, we modify the observation equation to a form that uses the nonlinear feature of JMM (we use only the muscle jacobian and the relative changes in muscle lengths), and does not use the absolute muscle lengths. Thus, the observation equation of EKF is changed as shown below.

$$\delta\boldsymbol{\theta}_k = \boldsymbol{\theta}_{k|k-1} - \boldsymbol{\theta}_{k-1|k-1} \tag{11}$$
$$\delta\boldsymbol{z}_k = G(\boldsymbol{\theta}_{k-1|k-1})\delta\boldsymbol{\theta}_k \tag{12}$$

The state equation is the same as Eq.7. In this equation, we observe not the absolute muscle lengths $\boldsymbol{z}_k$ but the relative changes in muscle lengths $\delta\boldsymbol{z}_k$. We can obtain $\delta\boldsymbol{z}_k$ by multiplying the muscle jacobian G and the difference between the previous and current joint angles. As described, by merely changing the observation equation, a joint angle estimation method that does not require the absolute muscle lengths is possible. Thus we do not need to calibrate muscle lengths in this formulation.

V. Experiments

As basic experiments, we validate that we can correctly construct the neck, scapula, and shoulder polyarticular-JMMs stated in Section III, and that we can estimate joint angles using polyarticular-JMMs in a simulation environment. Also, regarding the shoulder, we validate the effectiveness of the joint angle estimation method using the actual tendon-driven musculoskeletal humanoid, Kengoro. Next, we conduct experiments of joint angle estimation using only the relative changes in muscle lengths in a simulation environment. Also, we validate the effectiveness of the joint angle estimation method using the actual robot, Kengoro.

A. Joint Angle Estimation Using Polyarticular Joint-Muscle Mapping

First, we will summarize the computational cost required to construct the neck, scapula, and shoulder polyarticular-JMMs stated in Section III. We show the conditions and actual calculation time required to construct each polyarticular-JMM in Table I. The specifications of the machine used for this calculation are CPU:Intel(R) Core(TM) i7-5930K CPU @ 3.50GHz 12 Core, Memory:128GB. The data set was generated by setting N from 5 to 9 (we used 5 regarding joints having small movable ranges, and 9 regarding joints having large movable ranges). The number of the data set is N^D, as stated in Section III. The number of DOFs D and the number of muscles M are the same as stated in Section III. The data set construction and polynomial regression were completed in a practical amount of time, and so the approach in this study proved to be effective.

TABLE I. Computational cost to construct polyarticular-JMMs of upper limb of Kengoro (the neck, scapula, and shoulder).

	neck	scapula	shoulder
number of data (N^D)	1071875	2205000	2646000
DOFs (D)	8	8	8
number of muscles (M)	10	8	10
data set construction time [sec]	8579	16957	16840
polynomial regression time [sec]	5514	12885	25287

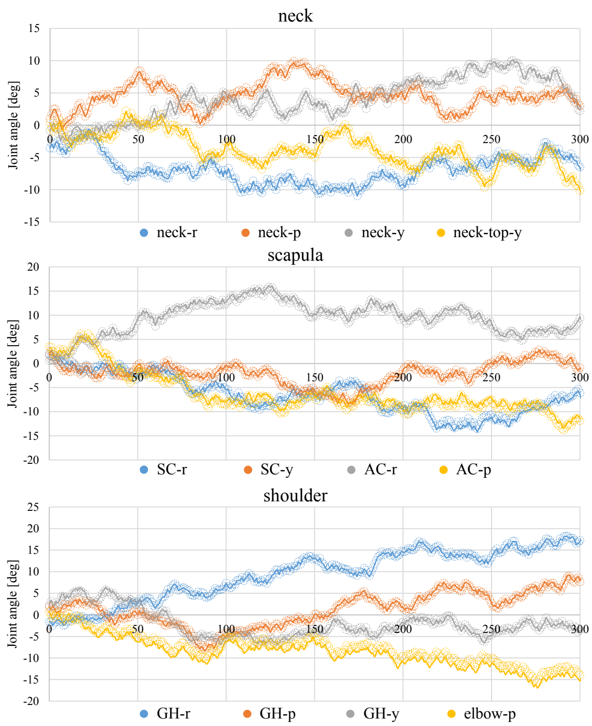

Fig. 6. The result of joint angle estimation of the neck, scapula, and shoulder using the geometric model. The actual joint angles (circle) and the estimated joint angles (line) are almost the same.

Second, we verified the joint angle estimation method using the polyarticular-JMMs. We used a geometric model in this experiment. We estimated joint angles from the displacement of muscle lengths when moving the joint angles of the model using random walk, and verified if the movement of the geometric model and estimated joint angles match. We show the result in Fig.6. The estimated joint angles match the actual movement, and we can see that the joint angle estimation method using polyarticular-JMM is feasible.

Third, we executed the joint angle estimation using the actual robot, Kengoro. We controlled Kengoro with constant tension control, moved Kengoro by applying force from the outside, and examined how the estimated joint angles changed. Because force is applied from the outside, it is easy for the estimation of joint angles to become incorrect by sudden noise, and we want to examine the robustness in this situation. We used the polyarticular-JMM obtained, stated above, and applied the conventional joint angle estimation method [7] to the right arm and the joint angle estimation formulated in this study, which considers the influence of polyarticular muscles, to the left arm. The conventional joint angle estimation is the estimation in which Eq.7–Eq.10 are not applied. First, we moved each elbow joint, and after that, moved both arms manually so that the joint angles will become almost the same. The result is shown in Fig.7. While the estimated joint angles vibrated regarding the conventional joint angle estimation in the right arm, the estimated joint angles were stable and the changes were smooth regarding the joint angle estimation method of this study in the left arm. In the experiment, because the conventional joint angle estimation method does not share duplicated joint angles among several polyarticular-JMMs, the value of $\boldsymbol{\theta}_n$ became strange and influenced the value of $\boldsymbol{\theta}_y$. Also, because the vibration is large, it is difficult to remove using a simple low-pass filter.

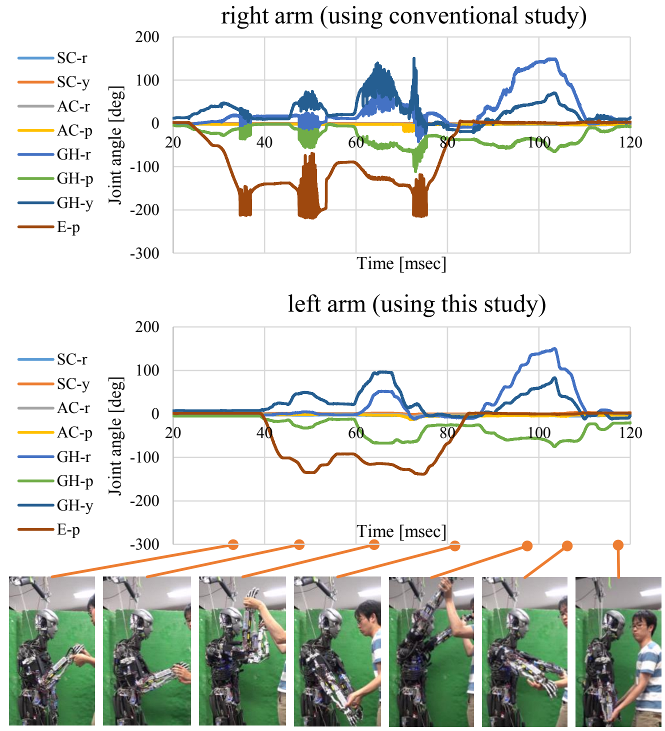

Fig. 7. The result of joint angle estimation of the neck, scapula, and shoulder using the actual robot, Kengoro.

B. Joint Angle Estimation Using Only the Relative Changes in Muscle Lengths

In this section, we used the formulation in Section III and Section IV, but unlike in the previous subsection, we used JMM expressed not by polynomials but by a NN. This is because JMM obtained using a geometric model is different from the JMM of the actual robot, and we need to make them more similar by the online learning of JMM [8] using the sensor information of the actual robot.

First, we conducted an experiment of relative-JAE stated in Section IV in a simulation environment using the geometric model of Kengoro. Regarding the shoulder and elbow (GH-r, GH-p, GH-y, E-p), we conducted this experiment in an

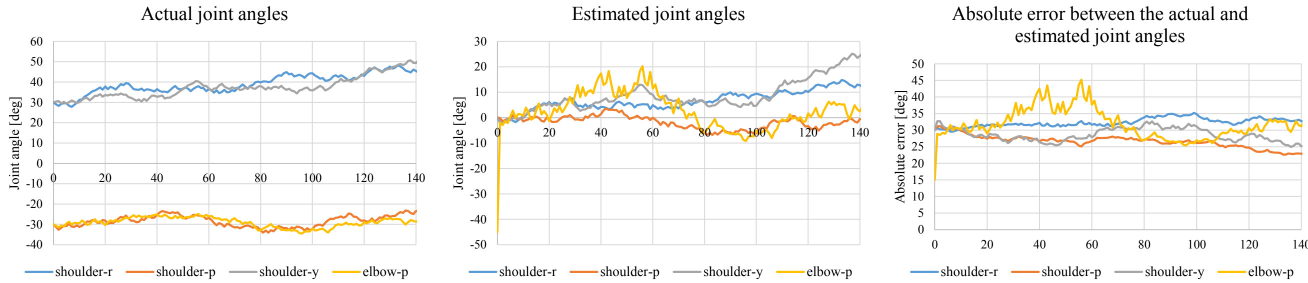

Fig. 8. The result of joint angle estimation of the shoulder using absolute muscle lengths in the geometric model.

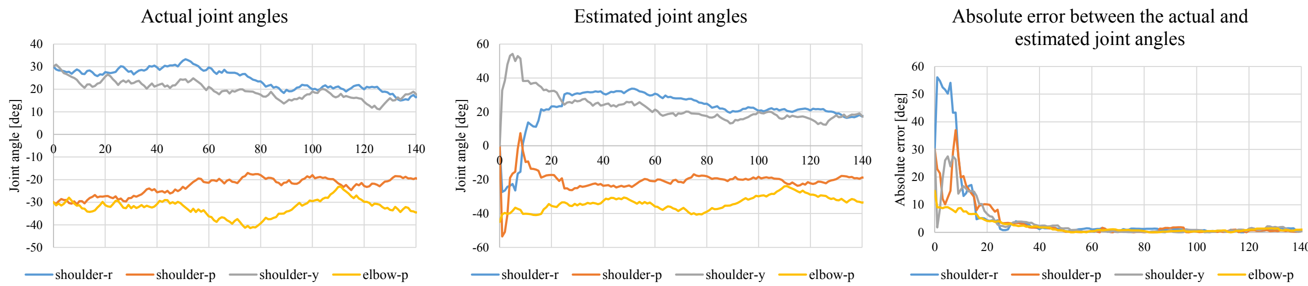

Fig. 9. The result of joint angle estimation of the shoulder using only relative changes in muscle lengths in the geometric model.

incorrectly calibrated situation in which the zero values of muscle lengths are calibrated when the joint angles are (30, -30, 30, -30) [deg]. We began the experiment when the initial estimated joint angles were (0, 0, 0, -45) [deg] (because the maximum joint angle of the elbow is 0 deg, we offsetted it to -45 deg), moved each joint angle of the geometric model by random walk, and verified the transition of the estimated joint angles. We show the result of the conventional joint angle estimation (Eq.1–Eq.2) in Fig.8. Because muscle lengths in JMM are 0 when joint angles are (0, 0, 0, 0) [deg], if joint angles at the muscle length calibration is misaligned at (30, -30, 30, -30) [deg], the estimated joint angles are unstable like in the center graph of Fig.8. Also, as shown in the right graph of Fig.8, the difference between the actual and estimated joint angles did not become smaller and the difference of about 30 deg did not change to a great extent. Next, we show the result of relative-JAE (Eq.12) in Fig.9. In this method, because joint angles are estimated by using only the relative changes in muscle lengths and the nonlinear feature of JMM, while the conventional joint angle estimation could not decrease the difference between the actual and estimated joint angles when the joint angles at the calibration were incorrect, this method can estimate joint angles correctly. As shown in the right graph of Fig.9, after random walk, the estimated joint angles became almost the same as the actual joint angles. Thus, we validated the effectiveness of this relative-JAE.

Second, we conducted experiments of relative-JAE using the actual robot Kengoro. We estimated only the elbow joint angle at first. The initial value of the estimated joint angle is -45 deg, and we incorrectly calibrated muscle lengths to 0 when the joint angle is about -100 deg. To evaluate the correctness of the estimated joint angles, we preserved the muscle lengths when we calibrated correctly at 0 deg, and we estimated the elbow joint angle using the conventional absolute-JAE at the same time as relative-JAE. If the two estimated joint angles are the same (we call the difference between the two estimated joint angles JAEs-error), we can verify the effectiveness of relative-JAE in the actual robot. The accuracy of the absolute-JAE is evaluated in [7], so we only need to evaluate the JAEs-error, and not the difference between the estimated joint angles by relative-JAE and the actual joint angles. The result is shown in the left of Fig.10. The JAEs-error became small to some extent (10–40 deg) during some exercises, but the result was not as good as in a simulation environment, because there is a model error between the actual robot and its geometric model. We show the result when we execute the antagonism updater of [8] for about 1 minute in the right of Fig.10 to correct antagonism of muscles. The JAEs-error became smaller (0–20 deg) than before learning.

Finally, we conducted experiments of relative-JAE using the shoulder of Kengoro. We calibrated muscle lengths to 0 in the situation that the shoulder joint angles (GH-r, GH-p, GH-y) were (40, -40, 40), and began the experiment by setting the initial estimated joint angles to (0, 0, 0) [deg]. We show the result in the left of Fig.11. The JAEs-error did not become smaller than in the elbow experiment, and the error fluctuated. Like the elbow experiment, we executed the antagonism updater of [8] and conducted the shoulder experiment again. The result is shown in the right of Fig.11. Although the JAEs-error became smaller, the fluctuation of error remained to some extent.

As a result, learning of JMM using the actual robot can make relative-JAE stable and precise. However, the model error between the actual robot and its geometric model could not be modified completely, so the results were not as satisfactory as those from the experiments in a simulation environment. Therefore, we need to consider a method to

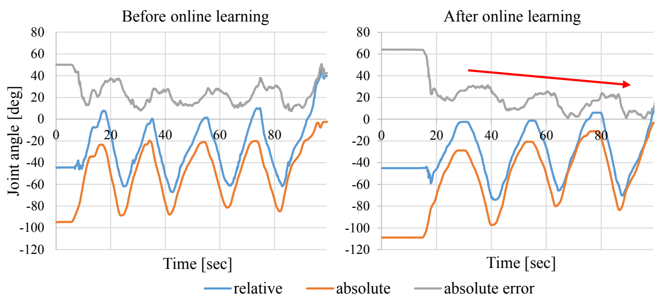

Fig. 10. The result of the elbow joint angle estimation using only relative changes in muscle lengths in the actual robot Kengoro. Left graph is before online learning [8], right graph is after online learning.

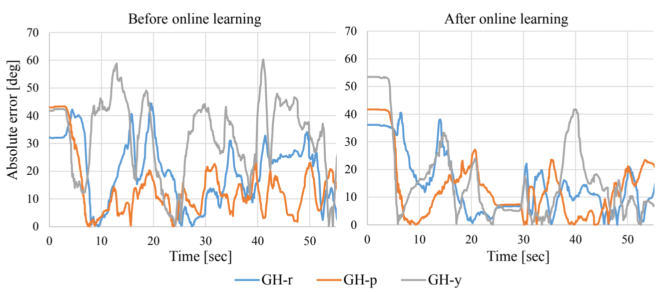

Fig. 11. The result of the shoulder joint angle estimation using only relative changes in muscle lengths in the actual robot Kengoro. Left graph is before online learning [8], right graph is after online learning.

make the JMM similar to the one in the actual robot and the formulation of a better joint angle estimation method.

VI. CONCLUSION

In this study, we discussed a method of joint angle estimation from muscle lengths in tendon-driven musculoskeletal humanoids, which include complex structures such as polyarticular muscles and the scapula, and an extended joint angle estimation method using only the relative changes in muscle lengths. Regarding the former, in previous joint angle estimation methods using data structures such as the data table, polynomials, and neural network, we stated that joint-muscle mapping (JMM), which expresses the nonlinear relationship between joint angles and muscle lengths, is important. Moreover, to construct the JMM, in terms of computational cost, we need to divide joints and muscles into several groups (polyarticular-JMMs) and formulate a joint angle estimation method that shares the estimated joint angles among groups while considering the effect of polyarticular muscles. Regarding the latter, we proposed a joint angle estimation method using only the relative changes in muscle lengths as an extension of the method using absolute muscle lengths. We considered the joint angle estimation of the current joint angles using the nonlinear feature of JMM by using not merely JMM but the differentiated value of JMM in the observation equation of EKF. Finally, we conducted experiments using these methods in simulation and actual environments, and verified the effectiveness. Regarding the latter, although the method functioned in a simulation environment or for the simple elbow joint of the actual robot, the estimated joint angles of the actual robot often diverged due to the model error between the actual robot and its geometric model, and so we need to consider a better learning framework to make the geometric model closer to the actual robot.

In future works, we would like to improve stability and precision of the joint angle estimation method using only the relative changes in muscle lengths. Also, although we divided joints and muscles into several groups manually in this study, we would like to do it automatically.

ACKNOWLEDGEMENT

The authors would like to thank Yuka Moriya (Ochanomizu University) for proofreading this manuscript.

REFERENCES

[1] Y. Nakanishi, S. Ohta, T. Shirai, Y. Asano, T. Kozuki, Y. Kakehashi, H. Mizoguchi, T. Kurotobi, Y. Motegi, K. Sasabuchi, J. Urata, K. Okada, I. Mizuuchi, and M. Inaba, "Design Approach of Biologically-Inspired Musculoskeletal Humanoids," *International Journal of Advanced Robotic Systems*, vol. 10, no. 4, p. 216, 2013.

[2] S. Wittmeier, C. Alessandro, N. Bascarevic, K. Dalamagkidis, D. Devereux, A. Diamond, M. Jäntsch, K. Jovanovic, R. Knight, H. G. Marques, P. Milosavljevic, B. Mitra, B. Svetozarevic, V. Potkonjak, R. Pfeifer, A. Knoll, and O. Holland, "Toward anthropomimetic robotics: Development, simulation, and control of a musculoskeletal torso," *Artificial Life*, vol. 19, no. 1, pp. 171–193, 2013.

[3] M. Jäntsch, S. Wittmeier, K. Dalamagkidis, A. Panos, F. Volkart, and A. Knoll, "Anthrob - A Printed Anthropomimetic Robot," in *Proceedings of the 2013 IEEE-RAS International Conference on Humanoid Robots*, 2013, pp. 342–347.

[4] Y. Asano, T. Kozuki, S. Ookubo, M. Kawamura, S. Nakashima, T. Katayama, Y. Iori, H. Toshinori, K. Kawaharazuka, S. Makino, Y. Kakiuchi, K. Okada, and M. Inaba, "Human Mimetic Musculoskeletal Humanoid Kengoro toward Real World Physically Interactive Actions," in *Proceedings of the 2016 IEEE-RAS International Conference on Humanoid Robots*, 2016, pp. 876–883.

[5] J. Urata, Y. Nakanishi, A. Miyadera, I. Mizuuchi, T. Yoshikai, and M. 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*, 2006, pp. 4428–4430.

[6] Y. Nakanishi, K. Hongo, I. Mizuuchi, and M. Inaba, "Joint proprioception acquisition strategy based on joints-muscles topological maps for musculoskeletal humanoids," in *Proceedings of The 2010 IEEE International Conference on Robotics and Automation*, 2010, pp. 1727–1732.

[7] S. Ookubo, Y. Asano, T. Kozuki, T. Shirai, K. Okada, and M. Inaba, "Learning nonlinear muscle-joint state mapping toward geometric model-free tendon driven musculoskeletal robots," in *Proceedings of the 2015 IEEE-RAS International Conference on Humanoid Robots*, 2015, pp. 765–770.

[8] K. Kawaharazuka, S. Makino, M. Kawamura, Y. Asano, K. Okada, and M. Inaba, "Online Learning of Joint-Muscle Mapping using Vision in Tendon-driven Musculoskeletal Humanoids," *IEEE Robotics and Automation Letters*, vol. 2, no. 4, pp. 2119–2126, 2018.

[9] M. V. Weghe, M. Rogers, M. Weissert, and Y. Matsuoka, "The ACT Hand: design of the skeletal structure," in *Proceedings of The 2004 IEEE International Conference on Robotics and Automation*, 2004, pp. 3375–3379.

[10] C. Paul, F. J. Valero-Cuevas, and H. Lipson, "Design and control of tensegrity robots for locomotion," *IEEE Transactions on Robotics*, vol. 22, no. 5, pp. 944–957, 2006.

[11] T. Yan, M. Cempini, C. M. Oddo, and N. Vitiello, "Review of assistive strategies in powered lower-limb orthoses and exoskeletons," *Robotics and Autonomous Systems*, vol. 64, pp. 120–136, 2015.

[12] I.A.KAPANDJI, *PHYSIOLOGIE ARTICULAIRE*, 6th ed. Ishiyaku Pub,Inc, 2010, vol. 1.