2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Biomimetic Control Scheme for Musculoskeletal Humanoids Based on Motor Directional Tuning in the Brain

Yasunori Toshimitsu[1], Kento Kawaharazuka[1], Kei Tsuzuki[1], Moritaka Onitsuka[1]
Manabu Nishiura[1], Yuya Koga[1], Yusuke Omura[1], Motoki Tomita[1]
Yuki Asano[1] , Kei Okada[1], Koji Kawasaki[2], and Masayuki Inaba[1]

Abstract— **In this research, we have taken a biomimetic approach to the control of musculoskeletal humanoids. A controller was designed based on the motor directional tuning phenomenon seen in the motor cortex of primates. Despite the simple implementation of the control scheme, complex coordinated movements such as reaching for target objects with its upper body was achieved, and is demonstrated in the accompanying video. The controller does not require an internal model, and instead constantly observes its body in relation to the external world to update motor commands. We claim that such an embodied approach to the control of musculoskeletal robots will be able to effectively take advantage of their complex bodies to achieve motion.**

I. INTRODUCTION

The aim of robotics research is to create physical autonomous agents that can move in the real world while accomplishing given tasks robustly. Biomimetic robots approach this goal by imitating bodily structures and behaviors of biological organisms, as they are already prime examples of autonomous systems operating in unstructured environments. Among them, musculoskeletal humanoids mimic the human musculoskeletal system, and are comprised of artificial muscle actuators spun across a passive joint structure.

Many of the control schemes successfully applied to whole-body musculoskeletal humanoids are model-based methods that are derived from concepts in robotics. Kawaharazuka et al. has used neural networks to model the relationship between joint space and muscle space [1], [2], and Jäntsch et al. has achieved torque control of musculoskeletal robots through tension-based control by computing the dynamics of the robot from a geometric model [3], [4]. Such model-based methods, while achieving joint-space control that can be used together with conventional robotic planners, are often set back by the nonlinear properties of the musculoskeletal robots which makes them hard to model.

On the other hand, model-free approaches have been researched as well. These tend to have simple rules to compute actuator output, and exploit the dynamic properties of the robots to achieve efficient control. Niiyama et al. achieved bipedal running of a musculoskeletal legged robot, by taking advantage of the spring-like properties of pneumatic muscles [5]. Martius et al. used a one-layer neural network that maps sensory input to actuator output, which produced emergent behavior on a musculoskeletal robotic arm [6].

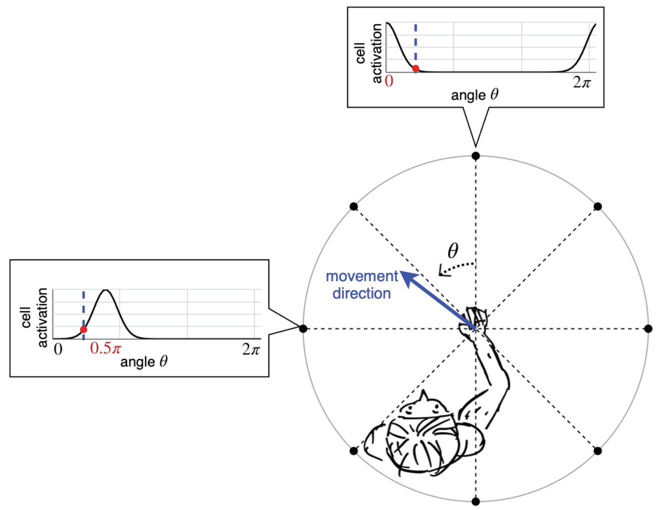


Fig. 1: Overview of motor directional tuning.

One of the proposed applications of musculoskeletal robots is their use as a human body simulator [7]. Musculoskeletal simulators such as OpenSim exist that can simulate dynamics of the human body, and are used in physiological research to analyze musculoskeletal dynamics and verify hypotheses regarding motor command generation [8].

Musculoskeletal robots can be regarded as a physical manifestation of such simulators, as its control program can be overwritten with any algorithm and various sensory information can be obtained from the distributed modules. Compared to computational simulators, motor control strategies applied to physical robots have some practical use; they can be used in real-world situations as a new control strategy for musculoskeletal robots.

This research attempts to mimic human motor control at an algorithmic level. Since the hardware of musculoskeletal humanoids are designed with a biomimetic approach, we can expect that a similar approach to their software can efficiently control these robots. However, compared to the musculoskeletal structure, much less is understood about the specific processes underlying human motor control. Thus, the physiological phenomenon must be appropriately abstracted

[1] The authors are with 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. [toshimitsu, kawaharazuka, tsuzuki, onitsuka, nishiura, koga, oomura, tomita, asano, k-okada, inaba]@jsk.t.u-tokyo.ac.jp
[2] The author is associated with TOYOTA MOTOR CORPORATION. koji_kawasaki@mail.toyota.co.jp

978-1-7281-6211-9/20/$31.00 ©2020 IEEE

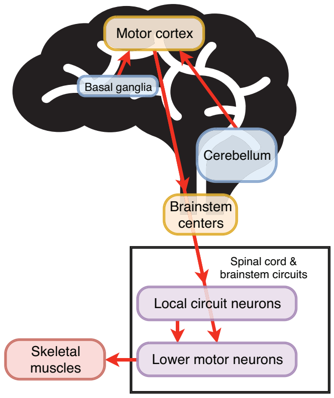


Fig. 2: Pathway of motor commands.

to an algorithm that can be run as a controller.

Similar approaches have been taken for musculoskeletal robots, to create controllers that can emulate various reflexes, such as the stretch reflex or reciprocal innervation between antagonistic muscles [9], [10]. However, these were low-level motor control strategies that can be seen in the spinal cord, and do not involve the higher-level regions such as the brain.

In this research, we have created a controller that emulates the phenomenon seen in the motor cortex called "motor directional tuning". This paper is structured as follows. In Section II, we will detail the original phenomenon seen in the motor cortex of primates. Then, in Section III, we will introduce the proposed controller that is based on the concept of motor directional tuning, and demonstrate its performance on reaching movements with a musculoskeletal arm. In Section V, we will apply the same controller to control the neck muscles of the musculoskeletal humanoid and combine it with an eye joint angle controller to achieve gaze control behavior.

II. MOTOR DIRECTIONAL TUNING SEEN IN THE MOTOR CORTEX

Motor directional tuning is a phenomenon seen in various parts of the brain regions involved in motor control, in which neurons are activated in an orderly fashion based on the direction of arm movement [11]. Fig. 1 illustrates this phenomenon. The motor cortical cells that exhibit this phenomenon each have a "preferred direction" that they are tuned to. This represents the arm movement direction in which the cell is most active. The cell's activity gradually decreases as the movement direction deviates from the cell's preferred direction. The activity of the cell plotted against the arm's movement direction is called a "directional tuning profile", and in Fig. 1 the directional tuning profiles of 2 cells are shown. It draws a bell-shaped curve with its peak at the cell's preferred direction. In the 1989 study by Georgopoulos et al. that first reported this phenomenon, hundreds of measured cells exhibited this behavior [12]. Motor directional tuning can be seen in arm movements in 3D space, and is stable across different movement amplitudes [13], [11]. This suggests that movement direction is one of the most dominant descriptors of arm movement, and motor cortical neurons are activated based on that variable.

Thus, we can expect that a cell tuned to a particular direction will activate a group of muscles that will move the hand in that direction. The activity of all the cells activated by motor directional tuning contribute to the final output, which are the neural signals sent to the muscles.

Here, it is important to recognize the role of the motor cortex in the motor system. The motor cortex exists in the mid-level of the motor control command pathway, as illustrated in Fig. 2. Therefore, the representations of output in the motor cortex are more abstracted than the individual muscle level. Instead, even a small stimulation in this area elicits activation of several muscles at once [14]. The information from the motor cortex are converted to commands for individual muscles as they pass through the motor command pathway.

The cerebellum and basal ganglia exist at a higher level than the motor cortex. The cerebellum is involved in sensory feedback and generation of coordinated movement, and the basal ganglia is involved in suppression of unwanted movements and initiation of movements.

At a lower level than the motor cortex, upper motor neurons in the brainstem centers use the motor commands from the motor cortex and combine it with sensory information such as vestibular, auditory, and visual information to keep balance of the body in response to internal and external disturbances. The motor commands from the motor cortex and brainstem centers travel through the neurons in the spinal cord and brainstem, where they finally are sent to muscles via the α motorneurons.

III. A CONTROLLER FOR MUSCULOSKELETAL HUMANOIDS BASED ON MOTOR DIRECTIONAL TUNING

Fig. 3 gives an overview of the proposed controller. Each muscle has a "muscle preferred direction", represented by a unit vector $\mathbf{c}_i$ for muscle i. The matrix C stores all the muscle preferred direction vectors used in the controller, as in Eq. 1. The number of muscles being controlled is n.

$$C = \begin{bmatrix} \mathbf{c}_1 & \mathbf{c}_2 & \dots & \mathbf{c}_n \end{bmatrix}^{\mathsf{T}} \quad (|\mathbf{c}_i| = 1,\ i\text{=1, 2, ..., n}) \tag{1}$$

In the actual phenomenon of motor directional tuning, the "preferred direction" is reprensented by cells in the motor cortex, and is more abstracted than the individual muscle level. However, in this proposed method, for simplicity, each preferred direction corresponds to a single muscle. Therefore, to differentiate from the original term, we call this the "muscle preferred direction". The process for measuring each muscle preferred direction for a musculoskeletal robot arm is described in Section IV-A.

The movement direction vector $\mathbf{m}$ is calculated according to the robot's task. The calculation process for an arm reaching movement is described here.

The movement direction vector $\mathbf{m}$ represents the intended direction and magnitude of movement of the arm. For

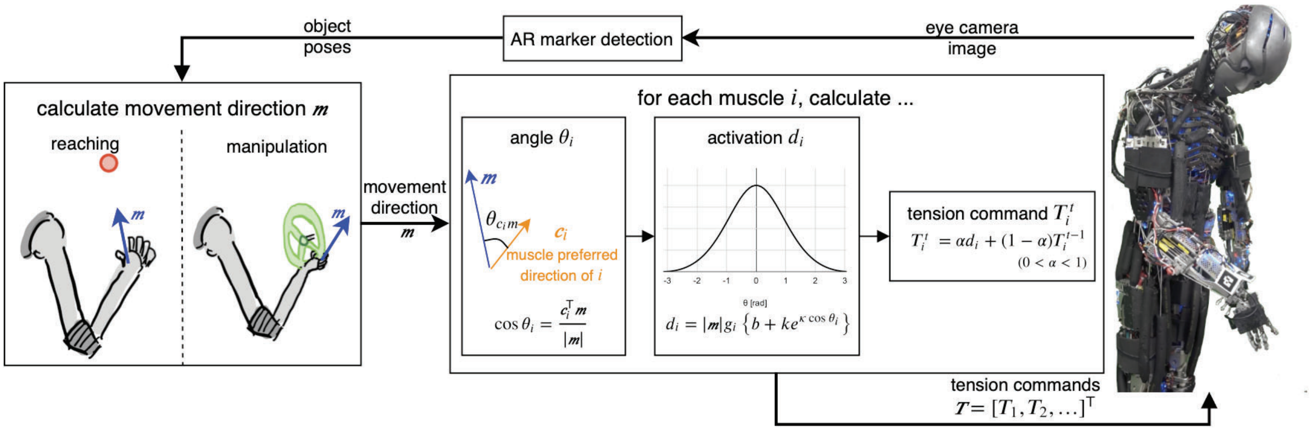


Fig. 3: Overview of proposed controller when applied to controlling arm movement.

reaching tasks, it was calculated by the following process.

$$\mathbf{v} = \mathbf{p}_{target} - \mathbf{p}_{hand}$$
$$\mathbf{m} = \begin{cases} \mathbf{v}/D_{thresh} & \text{if } |\mathbf{v}| < D_{thresh} \\ \mathbf{v}/|\mathbf{v}| & \text{otherwise} \end{cases} \tag{2}$$

Here, $|\cdot|$ expresses the L2 norm. $\mathbf{p}_{target}$ and $\mathbf{p}_{hand}$ are respectively the position of the target and hand in 3D space. $\mathbf{m}$ is parallel to the vector from the hand to the target $\mathbf{v}$, and is normalized by D_{thresh} with a maximum length of 1. D_{thresh} is determined based on the accuracy required in the reaching task. A smaller value improves precision, but when reaching from afar, could cause overshooting and oscillation around the target.

For other tasks, the movement direction vector can be calculated in a similar fashion. For example, in a handle rotation task as depicted in Fig. 3, the vector would be tangent to the handle. In this research, only the reaching task was performed.

The cosine of the angle between each muscle preferred direction $\mathbf{c}_i$ and movement direction $\mathbf{m}$ can be calculated from a simple matrix operation in Eq. 3, and then the directional activation d_i can be calculated from Eq. 4, which is based on the circular normal distribution used in research of motor directional tuning [11], [15].

$$\begin{aligned} \frac{C\mathbf{m}}{|\mathbf{m}|} &= \frac{\begin{bmatrix}\mathbf{c}_1^\mathsf{T}\mathbf{m} & \ldots & \mathbf{c}_n^\mathsf{T}\mathbf{m}\end{bmatrix}}{|\mathbf{m}|} \\ &= \begin{bmatrix}\cos\theta_1 & \ldots & \cos\theta_n\end{bmatrix} \end{aligned} \tag{3}$$

$$d_i = |\mathbf{m}| g_i \{b + k \exp(\kappa \cos\theta_i)\} \quad (i\text{=1, 2, ..., n}) \tag{4}$$

The parameters introduced in Eq. 4 will be explained.

κ determines the width of the tuning profile, as shown in Fig. 4. In this experiment, κ was set to 1 and the half width of the tuning profile is $128°$, wider than what has been physiologically observed. The median value of the half width of the directional tuning profile was $56°$ in the motor cortex [15]. This difference in width of the tuning profile is because the distribution of muscle preferred directions is sparse compared to that of preferred directions in the motor

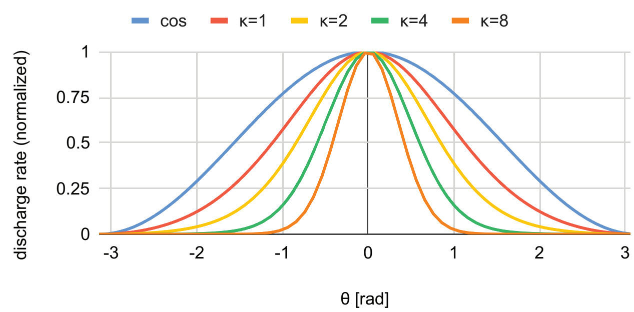


Fig. 4: Shape of directional tuning profile (described in Eq. 4) with different values of κ.

cortex. Whereas hundreds of cells have been observed to exhibit motor directional tuning, in this proposed controller, only 9 muscle preferred directions exist for arm movement. By having a wider directional tuning profile, we can ensure that muscles are activated across all movement directions.

b and k are set so that d_i takes a value between $0 \le d_i \le |m| g_i$ across all values of θ_i. They can be calculated as

$$\begin{aligned} k &= \frac{1}{e^\kappa - 1/e^\kappa} \\ b &= -\frac{k}{e^\kappa} \end{aligned} \tag{5}$$

g_i is the gain for each muscle, and when $|\mathbf{m}| \le 1$, defines the maximum tension output from that muscle. Since muscles closer to the center of the body tend to require more force output, g_i values for proximal muscles are set to be larger than that of distal muscles. Their values were determined experimentally through trial and error.

After the activation d_i is obtained, the tension command values to be sent to the robot is calculated by an exponentially weighted moving average function in Eq. 6. It acts as a low-pass filter, to prevent sudden changes in tension.

$$T_i^t = \alpha d_i + (1-\alpha) T_i^{t-1} \qquad (0 < \alpha < 1) \tag{6}$$

The value α in Eq. 6 is the smoothing factor of the exponentially weighted moving average function. It can be

TABLE I: Values of parameters used in experiment.

symbol	description	value
κ	parameter for tuning profile	1
k	parameter for tuning profile	0.425
b	parameter for tuning profile	-0.156
τ	time constant (arm, head)	(1.4, 0.7) [sec]
Δt	timestep (arm, head)	(0.25, 0.1) [sec]
α	smoothing factor (arm, head)	(0.16, 0.13)
T_{bias}	bias tension to keep wire taut	6 [N]
T_{max}	maximum tension limit for safety	400 [N]

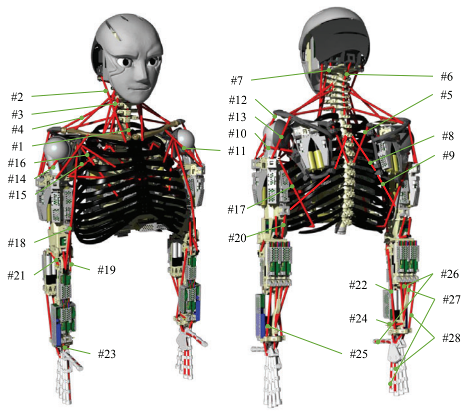


Fig. 5: Arrangement of upper muscles of Kengoro [1].

determined from a time constant τ and control cycle Δt.

$$\alpha = 1 - e^{-\Delta t/\tau} \tag{7}$$

The time constant τ is the time it takes for T_i^t to increase from zero to $1-1/e \approx 63.2\%$ of the original signal d_i, when d_i is kept constant.

Finally, the actual tension commands sent to the robot are processed with Eq. 8. To keep the muscle wire taut, the tension commands are kept above a constant offset T_{bias} and the command saturates at T_{max} to prevent large tension values from damaging the robot.

$$T_i^{\text{command}} = \min\{T_{bias} + \max\{T_i^t, 0\}, T_{max}\} \tag{8}$$

The values of each parameter which were used in the experiments are shown in Table II for g_i, and Table I for the rest.

IV. Implementation of the Proposed Controller on Musculoskeletal Humanoid Kengoro

The musculoskeletal tendon-driven humanoid Kengoro was used in this study. Kengoro has a musculoskeletal structure that closely resembles the human form, and is used as a platform for research on musculoskeletal robots [16]. Fig. 5 shows the muscle arrangement of the upper body of Kengoro. Table II lists the muscles used for each controller in this research. The arm controller uses 9 muscles, and the neck controller uses 4 muscles. The value g_i is the gain defined for each muscle and will be introduced later.

Fig. 6 shows the link and joint structure of Kengoro's upper limb. Among these, the shoulder joint and elbow joint was used in the arm controller, adding up to 4 degrees of freedom. The neck is comprised of 6 springs. The scapula and spine was not moved in this research.

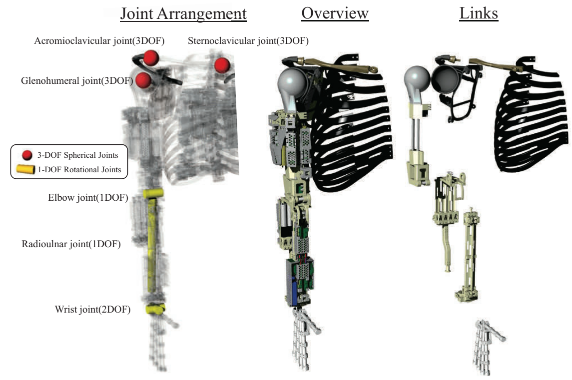


Fig. 6: Structure of Kengoro's upper limb [1].

TABLE II: Muscles used in each controller and their gains.

controller	muscles	g_i [N]
Arm	#10 *Infraspinatus*	450
	#11 *Deltoid* (front)	450
	#12 *Deltoid* (middle)	450
	#13 *Deltoid* (rear)	450
	#14 *Subscapularis*	450
	#15 *Pectoralis major*	300
	#18 *Biceps brachii*	150
	#19 *Brachialis*	100
	#20 *Anconeus*	70
Neck	#1 *Longus coli*	100
	#6 *Splenius capitis*	200
	#7 *Obliquus capitis superior* (left)	120
	#7 *Obliquus capitis superior* (right)	120

A. Process to obtain muscle preferred directions of arm muscles

The muscle preferred directions for each muscle $\mathbf{c}_i$ were obtained from a characterization process on the physical robot. They represent the direction of arm movement which the contraction of that muscle causes. By measuring them on a physical robot, we can capture characteristics not fully reproduced by a computational model.

The process is detailed in Algorithm 1. To obtain the muscle preferred direction for a muscle, tension is increased on just that muscle and the movement of the hand is observed by using the AR marker located on the forearm.

Fig. 7 shows the muscle preferred directions measured on the right arm.

B. Variance of muscle preferred directions across different poses

In our current implementation, the same muscle preferred directions are used across all poses. However, especially with the arm, where there is a complex musculoskeletal structure, the effect that a muscle exerts on the body differs depending on the pose of the robot. Thus, the muscle activation calculated from a constant muscle preferred direction may not cause the arm to move in the intended movement direction. For reaching movements, small differences in

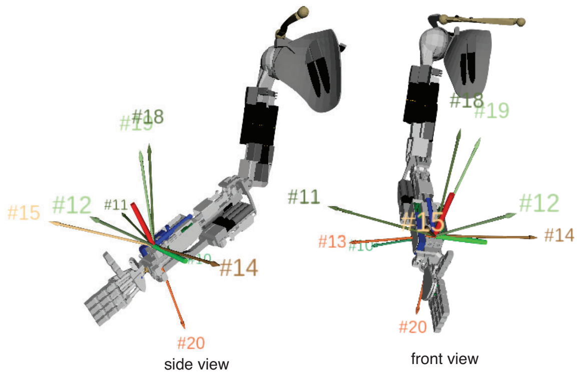
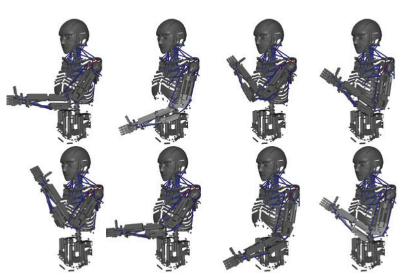


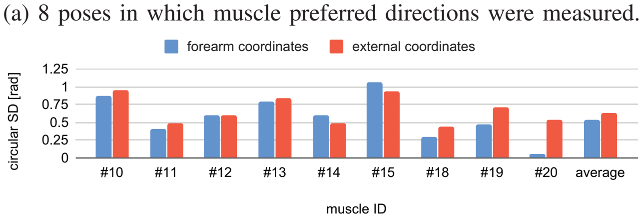
Fig. 7: Result of muscle preferred directions measured on right arm, displayed on 3D model for visibility.

Algorithm 1 Algorithm for calculating preferred directions.

```
robot is set to initial pose P by operator
for i = 1 to n do
    p_0 ←[position of hand]
    T_i = T_i+ΔT (keep tension of other muscles constant)
    sleep(2 [sec])
    p_1 ←[position of hand]
    v ← p_1 − p_0
    c_i ← v/|v|
    set robot to pose P by muscle length control
end for
C = [c_1  c_2 ...  c_n]^T
RETURN C
```



(a) 8 poses in which muscle preferred directions were measured.



(b) Circular standard deviation of muscle preferred directions.

Fig. 8: Circular standard deviation of muscle preferred directions measured in various poses.

intended movement direction and actual movement direction is not a concern. This is because as long as the controller can move the arm to bring the hand closer to the target, the hand can asymptotically reach the target. We have attempted to quantify this effect, by measuring muscle preferred directions in different poses and analyzing its variation.

A simulated model of Kengoro was used in this evaluation. This model is implemented as a tendon-driven robot on the MuJoCo physics engine [17]. The muscle preferred directions were measured in 8 different poses across the workspace as shown in Fig. 8 (a), and their circular standard deviations were calculated, as shown in Fig. 8 (b). Two types of coordinates were used to express the muscle preferred directions: forearm-fixed, and external coordinates. The average circular standard deviation was smaller for the forearm-fixed coordinates, at 0.53 [rad], or 30°. Thus, we can expect that for most of the workspace, the difference in intended movement direction and actual movement direction will stay within 90 °, and the hand can asymptotically reach the target.

In this research, a forearm-fixed coordinate was used. This also has the advantage of being computable from just the camera image. As long as both the hand and target AR markers are in sight, the movement direction vector can be calculated. If we were to calculate the vector in external coordinates, the orientation of the camera will also be necessary.

Next, we evaluate this controller for a reaching motion.

C. Comparison of Reaching Experiment with Model-Based Controller

A comparison experiment between the proposed controller and a controller based on a geometric musculoskeletal model was run in order to evaluate the proposed method. The task was to reach a visual target which was positioned (500, 50, 50) [mm] from the origin of Kengoro's coordinates (located at the base of the spine) with its hand, which was defined to be (10, -40, -150) [mm] from the AR marker on its forearm.

The geometric model-based method used for comparison is comprised of two steps. It first moves the robot to the desired position $\mathbf{x}_{hand}$ by solving the inverse kinematics of the arm. The muscle lengths are computed from the model and sent to the robot. In the equation below, f is the mapping between robot pose and muscle length.

$$\begin{aligned} \theta &= \mathit{inverse\_kinematics}(\mathbf{x}_{hand}) \\ \mathbf{L}_{muscle} &= f(\theta) \end{aligned} \tag{9}$$

Since the geometric model does not completely match the physical robot, even after this operation there will be an error $\Delta\mathbf{x}$ between the hand and target. The next step incrementally attempts to minimize this error in operational space. The robot was continuously controlled with the following simple update rule using the joint Jacobian J. The newly computed muscle lengths were continuously sent to the robot.

$$\begin{aligned} \Delta\theta &= rJ^{-1}\Delta\mathbf{x} \qquad (\, 0 < r \leq 1) \\ \theta &\leftarrow \theta + \Delta\theta \\ \mathbf{L}_{muscle} &= f(\theta) \end{aligned} \tag{10}$$

r is a coefficient which determines how fast the hand converges to the target position. In this experiment, r was set to 0.002 and the control cycle was 0.25 [sec].

For the proposed method, the D_{thresh} seen in Eq. 2 was first set to 0.4 [m], then was halved every time the arm motion stopped. This was in order to gradually improve the accuracy of the reaching motion, while preventing overshoots when too small a threshold is used from the beginning.

The results are shown in Fig. 9. It shows the robot at the end of the sequence, as well as graphs of muscle tension and distance between the hand and target. For the proposed method, the recognition result and computed movement direction vector $\mathbf{m}$ is shown with the robot, and for the comparison method, the internal geometric model is shown.

The model-based method achieved a higher precision. When the robot reached the target pose computed from inverse kinematics, it only had a precision of around 200 [mm]. After that, the incremental update through the use of joint Jacobian improved the accuracy, finally reaching closer than 7 [mm] to the target. On the other hand, the proposed method could only reach within 25 [mm] of the target. Most of this error was in the $+z$ direction (parallel to the forearm, in the proximal direction).

The reason for the remaining error in the proposed method can be attributed to the uneven distribution of the muscle preferred directions. As can be seen in Fig. 7, there is a sparse distribution of muscle preferred directions in the z direction (parallel to the forearm). Thus, there is weak activation of the muscles for this movement direction. This is a limitation of the current scheme for motor directional tuning based control, and we expect that by incorporating synergistic relations between muscles to the control scheme, precise movement in such directions will become possible.

As for muscle tensions, the proposed method had lower muscle tension overall, and no muscle increased over 150 [N]. On the other hand, in the model-based method, the tension for #19 *Brachialis* (elbow flexor) and #20 *Anconeus* (elbow extensor) increased as the incremental update progressed, reaching as high as 320 [N].

The high tension in the model-based method is due to the error between the model and robot. The model error is why the first pose solved by inverse kinematics does not actually reach the target. The update rule with joint Jacobian improves operational space accuracy on the real robot, but sacrifices model precision. In the end, there is a large pose difference between the model and robot, as can be seen in Fig. 9. This is especially prevalent on the elbow, and results in buildup of large internal tension. This phenomenon can be alleviated by improving the precision of the model, such as by using vision to update the joint-muscle mapping[1]. However, there will always be some error between the model and robot, and problems due to model error will persist.

In the proposed method, no internal model is used and muscle tension is calculated from the current relationship between the hand and target. Due to this, the muscle tension can be kept low compared to the other method. There is no internal force resulting from antagonistic muscle pairs, since only agonist muscles are activated from motor directional tuning. For example, from Fig. 7, it can be seen that the elbow flexors (#18 and #19) and extensor (#20) have muscle preferred directions in opposite directions. Thus, when the flexors are active, the extensor is inactive, and vice versa. As such, the simple activation rule from motor directional tuning can result in efficient usage of muscles.

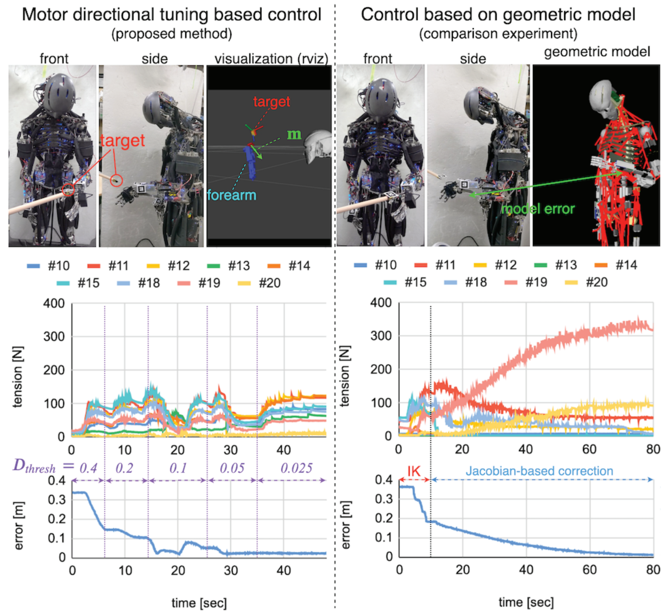


Fig. 9: Reaching experiment with proposed and model-based controller.

V. Gaze control using the proposed controller

The proposed controller was also apploed to control the neck muscles, combined with eye angle control with a more traditional controller. Human eyes have foveated vision, where there is a higher density of photoreceptors around the center of their field of vision. Such higher-resolution vision accounts for only a $1 \sim 2°$ visual field, and humans constantly move their eyes to take in visual information around themselves [14]. When combined with neck motion, a wide area can be observed without moving the torso.

Although the robot used in this research does not have foveated vision and has a constant resolution across its field of vision, it is still advantageous to implement such strategies since it can effectively increase the field of view of the camera. This is especially important for the proposed arm controller, since it must always have both markers on the arm and the target in sight to compute the muscle tension command. By coordinating the movement of the eye and neck, the robot can direct its gaze to a wider area, and therefore its arm can reach a wider workspace.

Two separate controllers were used to achieve this motion. The neck muscles were controlled by the motor directional tuning based controller, while the eyeball movements were controlled through conventional frame calculations. Each controller will be explained in the following subsections, followed by experimental data when these controllers were run at the same time.

A. Eye Angle Control

Algorithm 2 Process for eye control.

reset reference IMU pose
loop
  if object is outside threshold **then**
    follow object (duration 500 [ms])
    reset reference IMU pose
  else
    VOR (duration 5 [ms])
  end if
end loop

Algorithm 2 describes the process run on the eye. When the target object (in this case, the AR marker on the target object) is not in view, or is near the center of the field of vision, the **VOR** (vestibulo-ocular reflex) process is run. This reads the IMU values and sends joint angle commands to the eye so that the gaze direction is kept constant. It tries to keep the same gaze direction as when **reset reference IMU pose** is last called.

When the target object is outside a predefined threshold in the camera image, the **follow object** function is called and the eye is moved to re-center the object in the field of view.

B. Neck Muscle Control

The proposed controller based on motor directional tuning was used to control the neck muscles. They were controlled to align the head orientation with the eye gaze direction.

The movement direction vector $\mathbf{m}$ was calculated from the eye joint angles in a process similar to Eq. 2.

$$\mathbf{v}_{eye} = (\theta_y, \theta_p - 0.1[rad])^\mathsf{T}$$
$$\mathbf{m} = \begin{cases} \mathbf{v}_{eye}/0.17 & \text{if } |\mathbf{v}_{eye}| > 0.17 \\ \mathbf{v}_{eye}/|\mathbf{v}_{eye}| & \text{otherwise} \end{cases} \tag{11}$$

θ_y and θ_p are respectively the yaw and pitch angles of the eye, in radians. $\mathbf{m}$ is calculated as a vector from (θ_y, θ_p) to $(0, 0.1)$. The offset of 0.1 [rad] for the pitch angle is to counter the effect of gravity when looking down.

For the neck muscle controller, the muscle preferred directions were determined empirically from the musculoskeletal structure.

C. Gaze Control Experiment

The behavior of the controllers when they are run simultaneously is demonstrated in Fig. 10. First, the robot moves its eyes to an external visual target. Then, the neck is rotated to match the head orientation with the eyes. During this neck movement, gaze direction is kept constant due to the VOR.

The eye joint angle graph shows that after the eyes deviate from their initial position at (b), they return to the center position at (c). The neck muscles are activated by motor directional tuning to rotate and raise the head.

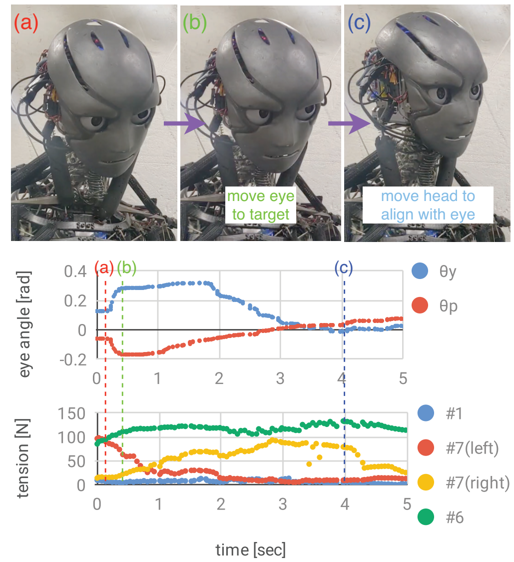


Fig. 10: Gaze control movement in reponse to a visual target.

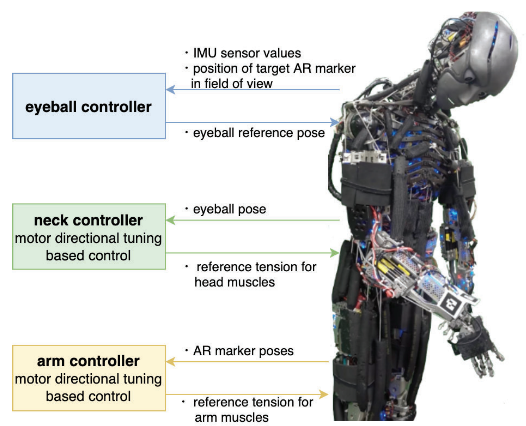


Fig. 11: I/O of each controller used in upper body control.

This coordinated behavior between the eye and neck muscles is not controlled by a central process, and is an emergent behavior that arises from two separate processes, coupled only by the eye joint angles.

D. Combining Gaze Control with Arm Control

Fig. 12 shows the behavior of the robot when the controllers for the arm, neck and eyes are run simultaneously. By controlling the gaze direction, the robot was able to reach a wider space than with a fixed camera angle. Each of the inputs and outputs of the three controllers are illustrated in Fig. 11. Although these controllers do not directly communicate with one another, coordinated motion was achieved by the upper body.

VI. CONCLUSION

In this research, we developed a biomimetic controller for musculoskeletal humanoids, and applied it to the control of the arm and neck. This controller uses a simple rule based on motor directional tuning to compute muscle tension

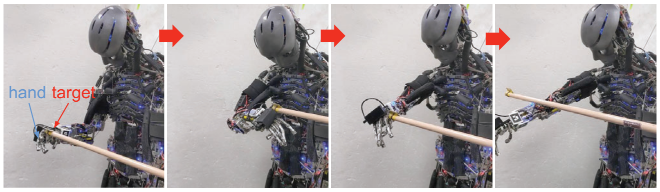


Fig. 12: Reaching for visual target with its upper body.

commands, and does not require an internal musculoskeletal model. Because it is not based on a physics model, we consider that this method can be easily applied to other types of musculoskeletal robots whose characteristics are difficult to model, such as pneumatic artificial muscles.

This controller maps movement direction given in operational space into tension commands, which are described in muscle space. It does not utilize the joint space, and parameters in this space (e.g. joint angle, joint torque) is not represented in the controller. Since humans are not considered to have organs that directly measure joint angle [18], [19], we believe that a biomimetic controller for musculoskeletal robots also should not utilize direct measurements of joint angles. Thus, the arm controller does not solve inverse kinematics to generate reaching motion, as conventional methods do. Instead, it constantly monitors the current relationship between its body and the target object, and activates muscles according to that relationship.

This controller takes advantage of the presence of the robot, and achieves motion through a simple mapping between input and output. By imitating the process seen in primates, we believe that the controller takes an embodied approach to the control of musculoskeletal robots.

Although this controller has realized efficient activation of muscles, it has failed to achieve comparable precision as model-based approaches. One possible improvement to this controller is to modify the mapping between each preferred direction and each muscle. Currently, their relationship is described with a simple one-on-one relationship, and is fixed even as the arm pose changes. By instead utilizing more complex mappings such as neural networks, we hope to create a motor directional tuning based controller that can take into account the synergistic relations between muscles, and update itself by constantly observing its input and output.

References

[1] 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. 3, no. 2, pp. 772–779, 2018.

[2] K. Kawaharazuka, K. Tsuzuki, S. Makino, M. Onitsuka, Y. Asano, K. Okada, K. Kawasaki, and M. Inaba, "Long-time self-body image acquisition and its application to the control of musculoskeletal structures," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 2965–2972, July 2019.

[3] M. Jäntsch, S. Wittmeier, K. Dalamagkidis, and A. Knoll, "Computed muscle control for an anthropomimetic elbow joint," in *Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 2192–2197.

[4] M. Kawamura, S. Ookubo, Y. Asano, T. Kozuki, K. Okada, and M. Inaba, "A Joint-Space Controller Based on Redundant Muscle Tension for Multiple DOF Joints in Musculoskeletal Humanoids," in *Proceedings of the 2016 IEEE-RAS International Conference on Humanoid Robots*, 2016, pp. 814–819.

[5] R. Niiyama, S. Nishikawa, and Y. Kuniyoshi, "Athlete Robot with applied human muscle activation patterns for bipedal running," in *Proceedings of the 2010 IEEE-RAS International Conference on Humanoid Robots*, 2010, pp. 498–503.

[6] G. Martius, R. Hostettler, A. Knoll, and R. Der, "Compliant control for soft robots: Emergent behavior of a tendon driven anthropomorphic arm," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2016, pp. 767–773.

[7] 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, pp. 216–228, 2013.

[8] A. Seth, J. L. Hicks, T. K. Uchida, A. Habib, C. L. Dembia, J. J. Dunne, C. F. Ong, M. S. DeMers, A. Rajagopal, M. Millard, S. R. Hamner, E. M. Arnold, J. R. Yong, S. K. Lakshmikanth, M. A. Sherman, J. P. Ku, and S. L. Delp, "Opensim: Simulating musculoskeletal dynamics and neuromuscular control to study human and animal movement," *PLOS Computational Biology*, vol. 14, no. 7, pp. 1–20, 07 2018. [Online]. Available: https://doi.org/10.1371/journal.pcbi.1006223

[9] X. Liu, A. Rosendo, S. Ikemoto, M. Shimizu, and K. Hosoda, "Robotic investigation on effect of stretch reflex and crossed inhibitory response on bipedal hopping," *Journal of The Royal Society Interface*, vol. 15, p. 20180024, 03 2018.

[10] K. Kawaharazuka, M. Kawamura, S. Makino, Y. Asano, K. Okada, and M. Inaba, "Antagonist Inhibition Control in Redundant Tendon-driven Structures Based on Human Reciprocal Innervation for Wide Range Limb Motion of Musculoskeletal Humanoids," *IEEE Robotics and Automation Letters*, vol. 2, no. 4, pp. 2119–2126, 2017.

[11] M. Mahan and A. Georgopoulos, "Motor directional tuning across brain areas: Directional resonance and the role of inhibition for directional accuracy," *Frontiers in neural circuits*, vol. 7, p. 92, 05 2013.

[12] A. P. Georgopoulos, J. F. Kalaska, R. Caminiti, and J. T. Massey, "On the relations between the direction of two-dimensional arm movements and cell discharge in primate motor cortex." *The Journal of neuroscience : the official journal of the Society for Neuroscience*, vol. 2 11, pp. 1527–37, 1982.

[13] A. B. e. a. Schwartz, "Primate motor cortex and free arm movements to visual targets in three-dimensional space. i. relations between single cell discharge and direction of movement." *The Journal of neuroscience : the official journal of the Society for Neuroscience*, vol. 8,8, pp. 2913–27., 1988.

[14] D. Purves, G. J. Augustine, D. Fitzpatrick, W. C. Hall, A.-S. LaMantia, J. O. McNamara, and S. M. Williams, Eds., *Neuroscience: Third Edition*. Sinauer Associates, Inc., 2004.

[15] B. Amirikian, A. Georgopoulos, and A. Georgopulos, "Directional tuning profiles of motor cortical cells," *Neuroscience research*, vol. 36, pp. 73–9, 02 2000.

[16] 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.

[17] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033, 2012.

[18] J. Taylor, "Proprioception," in *Encyclopedia of Neuroscience*, L. R. Squire, Ed. Oxford: Academic Press, 2009, pp. 1143 – 1149. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9780080450469019070

[19] V. G. Macefield, N. Hirokawa, and U. Windhorst, *Proprioception: Role of Joint Receptors*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 3315–3316. [Online]. Available: https://doi.org/10.1007/978-3-540-29678-2_4826