Artif Life Robotics (2013) 18:41–51
DOI 10.1007/s10015-013-0097-x

ORIGINAL ARTICLE

Muscle force distribution for adaptive control of a humanoid robot arm with redundant bi-articular and mono-articular muscle mechanism

Haiwei Dong · Nikolaos Mavridis

Received: 7 January 2013 / Accepted: 14 May 2013 / Published online: 30 May 2013

Abstract Robot arms driven by bi-articular and mono-articular muscles have numerous advantages. If one muscle is broken, the functionality of the arm is not influenced. In addition, each joint torque is distributed to numerous muscles, and thus the load of each muscle can be relatively small. This paper addresses the problem of muscle control for this kind of robot arm. A relatively mature control method (i.e. sliding mode control) was chosen to get joint torque first and then the joint torque was distributed to muscle forces. The muscle force was computed based on a Jacobian matrix between joint torque space and muscle force space. In addition, internal forces were used to optimize the computed muscle forces in the following manner: not only to make sure that each muscle force is in its force boundary, but also to make the muscles work in the middle of their working range, which is considered best in terms of fatigue. Besides, all the dynamic parameters were updated in real-time. Compared with previous work, a novel method was proposed to use prediction error to accelerate the convergence speed of parameter. We empirically evaluated our method for the case of bending-stretching movements. The results clearly illustrate the effectiveness of our method towards achieving the desired kinetic as well as load distribution characteristics.

Keywords Muscle cooperation · Redundancy · Internal force · Parameter adaptation

H. Dong (✉) · N. Mavridis
New York University AD, P.O. Box 129188, Abu Dhabi, UAE
e-mail: haiwei.dong@nyu.edu

N. Mavridis
e-mail: nikolaos.mavridis@nyu.edu

1 Introduction

1.1 Background

Traditional robot arms are driven by motors. Each motor drives a specific joint, corresponding to an independent degree of freedom (DOF). There are two main problems in this approach. First, if one motor is broken, the DOF corresponding to this motor will be lost. Second, the motor near the base link always needs more power, leading to a requirement for very heavy motors near the base. But let us compare this to the human motor system: as there are many muscles driving one joint, the total joint torque is distributed to every muscle. Thus, each muscle only carries a relatively small load. Furthermore, if one muscle is broken, the function of rotation for the corresponding joint does not change. Thus, recently such driving systems for robots inspired by muscles have started to be explored, creating a new research direction. For example, there are a number of papers focusing on bionic arms [1–4]. In our work, we propose a muscle control method for humanoid robot arm driven by bi-articular muscles. Here, the control method is designed to be adaptive, i.e. robust to perturbations and disturbances from environment. In addition, the muscle forces have to satisfy the boundary force limit. Last, but quite importantly, the method has to be efficient enough for practical application.

To get insight into this muscle control problem, we can consider it through viewpoints arising from two research fields: human motor control and robotics. In human motor control, the human body is usually considered as a multi-link rigid body with numerous joints. By adding muscles and tensors, the human body moves according to control signals arising from the neural system. Such a kind of system is usually termed a neural-skeleton muscle system

[5, 6] and the muscle control is described as human motor control [7]. On the other hand, in robotics, there is also a related research field focusing on controlling multi-link rigid bodies [8–10].

In our research, we face two problems: redundancy and adaptivity. For redundancy, there exist two types of redundancies here. Type I Redundancy is between end effector position and joints. Taking the robot arm as an example, the number of degrees of freedom (DOF) of the end effector position is 3, i.e., the end effector can move in 3D space. Nevertheless, the number of joints in the arm can be more than 3, indicating the fact that for the same end effector position, there exist many limb joint configurations. The other type of redundancy, Type II Redundancy, is between joint torque and muscle force. In the human motor system, there are many more muscles as compared to the minimum number required to generate the full range of required movement [11, 12]. Towards generating a certain desired joint torque, there is a lot of flexibility in distributing load among the cooperating muscles. For example, if we ask a subject to keep a certain gesture, when the subject is under nervous status, the muscles are tense. Both agonist muscles and antagonist muscles output large force. At this time, the body's impedance status is in a self-protection mode, and the damping and viscosity coefficients are correspondingly large [13, 14]. Conversely, when the subject is under relaxation status, the muscle force, damping and viscosity coefficients are small. The above two extreme statuses correspond to the same zero joint torque with different muscle force distributions.

Ability for adaptation is also a very important issue, especially for humanoid robots. As humanoid robots are usually tailored towards significant human–robot interaction, often their working environment is mainly a human environment, which is very complex and always accompanied with various kinds of uncertainties. Furthermore, even if we do not consider the environmental uncertainty, and we focus on the robot itself, we still often have robot model errors. Thus, in order to be able to create humanoids that perform adequately under such conditions, it is necessary to propose methods that have the ability to perform real-time adaptation, for counteracting such uncertainties.

1.2 Related work

Let us start by discussing relevant work in human–robot control, and then moving over to relevant work in robotics. In human motor control, two of the main research directions are those focusing towards redundancy solutions and ability for adaptation. Let us start with the first: redundancy solution. The basic method utilized is nonlinear optimization [15–18]. There have been many successful applications. Hogan proposed a voluntary movement principle using dynamic optimization which minimizes the rate of acceleration change of the limb [19]. Anderson et al. [15] used dynamic optimization of minimum metabolic energy expenditure to solve the motion control of walking. Manal et al. [11] designed a nonlinear optimal controller to develop a real-time EMG-driven virtual arm. Neptune [16] evaluated different multivariate optimization methods in pedaling.

The optimization method in human motor control includes two different kinds of methods. One is forward dynamics method which uses muscle excitations as the inputs to calculating the corresponding body motions [12, 20]. It is a novel research in forward dynamics that Anderson et al. [21] uses a parallel computer to calculate the derivatives of the cost function and the constraints with each control variable. The other is inverse dynamics method [18]. Noninvasive measurement of body motions (position, velocity, and acceleration of each segment) and external forces are used as inputs to calculate muscle forces. Two commercial software packages are famous based on these two methods: AnyBody Modeling System as inverse dynamics method [22] and OpenSim as forward dynamics method [23].

As mentioned above, a second main research direction focuses on adaptation ability. Although adaptation ability based on bio-feedback has not been fully understood in human motor control, right now the basic comprehension is a combination control scheme of feed-forward control and feedback control [24]. The feed-forward control corresponds to the inverse dynamics [25–27]. Specifically, the inverse dynamics are learnt (i.e., estimated) by nervous system. Then, the feedback control uses this inverse dynamics to provide a basic control efficiently. The feedback control deals with environment disturbance, load change, and learning error of inverse dynamics, etc. It is usually considered as a Visual Servo Feedback System [28].

Now let us move from human motor control to robotics. In robotics, modeling and control of multi-link rigid body (especially manipulator control) have been studied for years [10, 29–31]. Here, the redundancy problem is usually considered from the viewpoint of dynamic control [1, 32]. For Type I Redundancy problem, considering different optimization criterion or restricted conditions, many methods have been proposed. Yoshikawa [29] proposed a manipulability measure, by minimizing which the arm is kept away from singularities. Maciejewski et al. [30] used null-space vector to aid obstacle avoidance. For Type II Redundancy problem, there have been researches on building bionic robots to mimic human's movement system. Klug et al. [2] developed a 3 DOF bionic robot arm which is controlled by a PD controller with feed-forward compensation. The trajectory of the arm is optimized and adjusted for a time and energy-optimal motion [33]. Potkonjak et al. [3] built a humanoid robot with antagonistic drives whose controller is designed by H_∞ loop shaping. Tahara [4] proposed a simple sensor-motor control scheme as internal force and simulated the overall stability.

The initial research on adaptation ability in robotics comes from online system identification. The objective is to estimate system structure and parameters in real-time. Here, the system is considered as a black box or gray box. By stimulating the system and creating relation between the inputs and outputs, the parameters of the system can be adjusted online [34]. Based on this identification idea, many adaptive methods came out, such as robust control [35], adaptive feedback control [36], neurofuzzy adaptive control [37], etc. However, the adaptation ability problem has not been considered in arm control by bi-articular muscles.

1.3 Our solution

In this paper, we took into account the previous research in human motor control and robotics. One of our initial observations was that the optimization-series methods are difficult to use in practical applications because of their relative inefficiency and difficulty in dealing with. It is also difficult to directly propose a muscle control method for the robot arm as it is hard to decouple the two redundancies. Therefore, we use a relatively mature control method (i.e. sliding mode control) to get joint torque first and then distribute the joint torque to muscle forces. The muscle force was computed based on a Jacobian matrix between joint torque space and muscle force space. In addition, internal forces were used to optimize the computed muscle forces in the following manner: not only to make sure that each muscle force is in its force boundary, but also to make the muscles work in the middle of their working range, which is considered best in terms of fatigue. Besides, all the dynamic parameters were updated in real-time. A novel method was proposed to use prediction error to accelerate the convergence speed of parameter. We empirically evaluated our method for the case of bending-stretching movements. The results clearly illustrate the effectiveness of our method towards achieving the desired kinetic as well as load distribution characteristics.

2 Modeling arm with muscles

2.1 Arm model

We built a 2-dimensional model of the arm in the horizontal plane based on the upper limb structure of a digital human. The model includes six muscles (shown as 1–6) and two degrees of freedom (shoulder flexion–extension and elbow flexion–extension). The range of shoulder angle is from −20 to 100°, and the range of the elbow is from 0 to 170°. Four of the muscles are mono-articular, and two are bi-articular where 1 and 2 cross the shoulder joint; 3 and 4 cross the elbow joint; 5 and 6 cross both joints (Fig. 1).

Considering the arm (including upper arm and lower arm) as a planar, two-link, articulated rigid object, the position of hand can be derived by a 2-vector q of two angles. The input is a 6-vector F_m of muscle forces. The dynamics of the rigid object is strongly nonlinear. Using the Lagrangian equations in classical dynamics, we get the dynamic equations of the ideal upper limb model

$$\begin{bmatrix} H_{11}(t) & H_{12}(t) \\ H_{21}(t) & H_{22}(t) \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} + \begin{bmatrix} C_{11}(t) & C_{12}(t) \\ C_{21}(t) & C_{22}(t) \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} + \begin{bmatrix} G_1(t) \\ G_2(t) \end{bmatrix} = \begin{bmatrix} \tau_1(t) \\ \tau_2(t) \end{bmatrix} \qquad (1)$$

or abbreviated as

$$H(t)\ddot{q} + C(t)\dot{q} + G(t) = \tau \qquad (2)$$

with $q = [q_1 \quad q_2]^{\mathrm{T}} = [\theta_1 \quad \theta_2]^{\mathrm{T}}$ being the two joint angles. $\tau = [\tau_1 \quad \tau_2]^{\mathrm{T}} = f(F_m)$ is a function of muscle force F_m

$$F_m = [F_{m,1} \quad F_{m,2} \quad F_{m,3} \quad F_{m,4} \quad F_{m,5} \quad F_{m,6}]^{\mathrm{T}} \qquad (3)$$

$H(q,t)$ is inertia matrix containing information with regard to the instantaneous mass distribution. $C(q,\dot{q},t)$ is centripetal and coriolis torques representing the moments

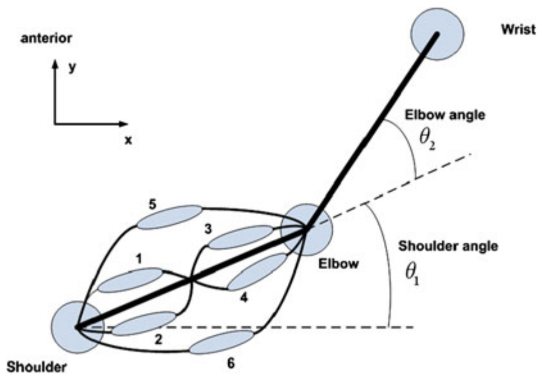

Fig. 1 Humanoid robot arm model. The arm has two degrees of freedom and it can rotate around the shoulder angle and elbow angle in the anterior plane

of centrifugal forces. $G(q,t)$ is gravitational torques changing with the posture configuration of the arm.

$$
\begin{aligned}
&H_{11} = J_1 + J_2 + m_2 d_1^2 + 2m_2 d_1 c_2 \cos(q_2)\\
&H_{12} = H_{21} = J_2 + m_2 d_1 c_2 \cos(q_2)\\
&H_{22} = J_2\\
&C_{11} = -2m_2 d_1 c_2 \sin(q_2)\dot{q}_2\\
&C_{12} = -m_2 d_1 c_2 \sin(q_2)\dot{q}_2\\
&C_{21} = m_2 d_1 c_2 \sin(q_2)\dot{q}_1\\
&\mathrm{C}_{22} = 0\\
&G_1 = g(m_1 c_1 + m_2 d_1)\cos(q_1) + g m_2 c_2 \cos(q_1 + q_2)\\
&G_2 = g m_2 c_2 \cos(q_1 + q_2),
\end{aligned}
\tag{4}
$$

where g is the acceleration of gravity. c_i is the distance from the center of a joint i to the center of the gravity point of link i. d_i is the length of link i. $J_i = m_i d_i^2 + I_i$ where I_i is the moment of inertia about axis through the center of mass of link i.

2.2 Model with estimated parameters

In our research, we consider the arm model in Eq. (1) is influenced by disturbances and perturbations from the environment. Hence, we used an estimated arm model to control, which is written as

$$
\begin{bmatrix} \hat{H}_{11}(t) & \hat{H}_{12}(t) \\ \hat{H}_{21}(t) & \hat{H}_{22}(t) \end{bmatrix}
\begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix}
+
\begin{bmatrix} \hat{C}_{11}(t) & \hat{C}_{12}(t) \\ \hat{C}_{21}(t) & \hat{C}_{22}(t) \end{bmatrix}
\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix}
+
\begin{bmatrix} \hat{G}_1(t) \\ \hat{G}_2(t) \end{bmatrix}
=
\begin{bmatrix} \hat{\tau}_1(t) \\ \hat{\tau}_2(t) \end{bmatrix}
\tag{5}
$$

or abbreviated as

$$\hat{H}(t)\ddot{q} + \hat{C}(t)\dot{q} + \hat{G}(t) = \hat{\tau}, \tag{6}$$

where $\hat{\cdot}$ means estimated value of $(\cdot)$. The connection part between ideal model (Eq. 1) and estimated model (Eq. 5) is that we choose $\tau = \hat{\tau}$. Below, we use estimated model (5) to generate torque for real system for control. In addition, the parameter adaptation updates the estimated parameters $\hat{H}$, $\hat{C}$ and $\hat{G}$ in real time.

2.3 Dynamic parameters definition

For the convenience of following derivation, we define the actual and estimated arm parameter vector

$$P = \begin{bmatrix} P_H^{\mathrm{T}} & P_C^{\mathrm{T}} & P_G^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}, \quad \hat{P} = \begin{bmatrix} \hat{P}_H^{\mathrm{T}} & \hat{P}_C^{\mathrm{T}} & \hat{P}_G^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}, \tag{7}$$

where

$$
\begin{aligned}
&P_H = \begin{bmatrix} H_{11} \\ H_{12} \\ H_{21} \\ H_{22} \end{bmatrix}, \quad P_C = \begin{bmatrix} C_{11} \\ C_{12} \\ C_{21} \\ C_{22} \end{bmatrix}, \quad P_G = \begin{bmatrix} G_1 \\ G_2 \end{bmatrix},\\
&\hat{P}_H = \begin{bmatrix} \hat{H}_{11} \\ \hat{H}_{12} \\ \hat{H}_{21} \\ \hat{H}_{22} \end{bmatrix}, \quad \hat{P}_C = \begin{bmatrix} \hat{C}_{11} \\ \hat{C}_{12} \\ \hat{C}_{21} \\ \hat{C}_{22} \end{bmatrix}, \quad \hat{P}_G = \begin{bmatrix} \hat{G}_1 \\ \hat{G}_2 \end{bmatrix}
\end{aligned}
\tag{8}
$$

then the estimation error vector can be defined as

$$\tilde{P} = \hat{P} - P = \begin{bmatrix} \tilde{P}_H^{\mathrm{T}} & \tilde{P}_C^{\mathrm{T}} & \tilde{P}_G^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}. \tag{9}$$

3 Joint torque computation

3.1 Torque control method

Sliding mode control is used to control the posture of the arm [36]. A 2-vector q_d is the desired states. Define a sliding mode term s as

$$s = \dot{\tilde{q}} + \Lambda\tilde{q} = (\dot{q} - \dot{q}_d) + \Lambda(q - q_d) \tag{10}$$

where Λ is a positive diagonal matrix. Defining the reference velocity $\dot{q}_{\mathrm{r}}$ and reference acceleration $\ddot{q}_{\mathrm{r}}$ as

$$
\begin{aligned}
&\dot{q}_{\mathrm{r}} = \dot{q} - s\\
&\ddot{q}_{\mathrm{r}} = \ddot{q} - \dot{s}
\end{aligned}
\tag{11}
$$

then we choose the control method as

$$\tau = \hat{H}(q)\ddot{q}_r + \hat{C}(q,\dot{q})\dot{q}_r + \hat{G} - K\mathrm{sgn}(\mathrm{s}), \tag{12}$$

where K is convergence parameter which is a diagonal matrix. The proof of sliding mode control is in [38].

3.2 Parameter adaptation method

To accelerate the parameter update speed, we use two error sources to update estimated parameters. The first source is tracking error. We chose the parameter adaptation method based on tracking error as

$$\dot{\hat{P}}_{\mathrm{tra}} = -\Gamma^{-1}\begin{bmatrix} s_1\ddot{q}_r^{\mathrm{T}} & s_2\ddot{q}_r^{\mathrm{T}} & s_1\dot{q}_r^{\mathrm{T}} & s_2\dot{q}_r^{\mathrm{T}} & s_1 & s_2 \end{bmatrix}^{\mathrm{T}}, \tag{13}$$

where Γ is adaptation parameter which is a diagonal matrix. The derivation of this parameter adaptation method is based on the sliding mode control method, which is proved in [39].

On the other side, the dynamic equation (Eq. 1) can be written in the form

$$\tau(t) = \begin{bmatrix} \ddot{q}_1 & 0 & \ddot{q}_1 & 0 & \dot{q}_1 & 0 & \dot{q}_1 & 0 & 1 & 0 \\ 0 & \ddot{q}_2 & 0 & \ddot{q}_2 & 0 & \dot{q}_2 & 0 & \dot{q}_2 & 0 & 1 \end{bmatrix} \begin{bmatrix} P_H \\ P_C \\ P_G \end{bmatrix}. \tag{14}$$

To avoid the acceleration terms in Eq. (14), we use filtering technique. Specifically, by multiplying both sides of Eq. (14) with $\mathrm{e}^{-\lambda(t-r)}$ and integrating it, we can get

$$\int_0^t \mathrm{e}^{-\lambda(t-r)}\tau(r)\mathrm{d}r = \int_0^t \mathrm{e}^{-\lambda(t-r)} \times \begin{bmatrix} \ddot{q}_1 & 0 & \ddot{q}_1 & 0 & \dot{q}_1 & 0 & \dot{q}_1 & 0 & 1 & 0 \\ 0 & \ddot{q}_2 & 0 & \ddot{q}_2 & 0 & \dot{q}_2 & 0 & \dot{q}_2 & 0 & 1 \end{bmatrix} \mathrm{d}r \begin{bmatrix} P_H \\ P_C \\ P_G \end{bmatrix}, \tag{15}$$

where λ and r are positive numbers. Using partial integration, the acceleration terms on the right side can be written as

$$\int_0^t \mathrm{e}^{-\lambda(t-r)} \begin{bmatrix} \ddot{q}_1 & 0 \\ 0 & \ddot{q}_2 \end{bmatrix} \mathrm{d}r = \mathrm{e}^{-\lambda(t-r)} \begin{bmatrix} \dot{q}_1 & 0 \\ 0 & \dot{q}_2 \end{bmatrix} \Bigg|_0^t - \int_0^t \frac{\mathrm{d}}{\mathrm{d}r}\left(\mathrm{e}^{-\lambda(t-r)} \begin{bmatrix} \dot{q}_1 & 0 \\ 0 & \dot{q}_2 \end{bmatrix} \right) \mathrm{d}r. \tag{16}$$

Therefore, Eq. (14) can be written in the form

$$\tau(t) = S(t, \dot{q}, q)P \tag{17}$$

From this equation, τ is the "output" of the system. S is a signal matrix. P is a vector of real parameters. We can predict the value of the output τ based on the parameter estimation, i.e.

$$\hat{\tau} = S\hat{P} \tag{18}$$

Then, the prediction error e can be defined as

$$e = \hat{\tau} - \tau = S\hat{P} - SP = S\tilde{P}. \tag{19}$$

According to it, we can get the parameter adaptation method based on prediction error, i.e.

$$\begin{aligned} \dot{\hat{P}}_{\mathrm{pre}} &= -\Xi \frac{\partial (e^{\mathrm{T}} e)}{\partial \hat{P}} = -\Xi \frac{\partial \left(\left(S\hat{P} - SP\right)^{\mathrm{T}} \left(S\hat{P} - SP\right) \right)}{\partial \hat{P}} \\ &= -2\Xi S^{\mathrm{T}} \left(S\hat{P} - SP\right) = -2\Xi S^{\mathrm{T}} e = -2\Xi S^{\mathrm{T}} (\hat{\tau} - \tau), \end{aligned} \tag{20}$$

where Ξ is a diagonal coefficient matrix. If we consider the parameters change much slower with respect to the parameter identification, from Eq. (20), we can get

$$\dot{\tilde{P}} = \dot{\hat{P}} - \dot{P} = -2\Xi S^{\mathrm{T}} S\tilde{P} \tag{21}$$

Here, we choose a Lyapunov function candidate

$$V(t) = \frac{1}{4}\tilde{P}^{\mathrm{T}}\tilde{P} \tag{22}$$

then the derivative of $V(t)$ is

$$\dot{V}(t) = \frac{1}{2}\tilde{P}^{\mathrm{T}}\dot{\tilde{P}} = \frac{1}{2}\tilde{P}^{\mathrm{T}}\left(-2\Xi S^{\mathrm{T}} S\tilde{P}\right) = -\Xi \left(S\tilde{P}\right)^{\mathrm{T}} \left(S\tilde{P}\right) \leq 0 \tag{23}$$

which means the parameter estimation converges to real values. Therefore, according to Eqs. (13) and (20), the overall adaptation law is

$$\begin{aligned} \dot{\hat{P}} &= \dot{\hat{P}}_{\mathrm{tra}} + \dot{\hat{P}}_{\mathrm{pre}} \\ &= -\Gamma^{-1} \left[s_1 \ddot{q}_r^{\mathrm{T}} \quad s_2 \ddot{q}_r^{\mathrm{T}} \quad s_1 \dot{q}_r^{\mathrm{T}} \quad s_2 \dot{q}_r^{\mathrm{T}} \quad s_1 \quad s_2 \right]^{\mathrm{T}} \\ &\quad - 2\Xi S^{\mathrm{T}} (\hat{\tau} - \tau). \end{aligned} \tag{24}$$

4 Muscle force distribution

4.1 Jacobian matrix between joint and muscle space

The coordinate system of the robot arm is shown in Fig. 2 where we define $a_{ij} (1 \leq i \leq 6, 1 \leq j \leq 2)$ as the distance between the muscle endpoint and center of its adjacent joint. Define $l_k (1 \leq k \leq 6)$ as the kth muscle length.

According to Sines Law and Cosines Law, we can get

$$\begin{aligned} l_1^2 &= a_{11}^2 + a_{12}^2 - 2a_{11}a_{12}\cos(\pi - \theta_1) \\ l_2^2 &= a_{21}^2 + a_{22}^2 - 2a_{21}a_{22}\cos(\theta_1) \\ l_3^2 &= a_{31}^2 + a_{32}^2 - 2a_{31}a_{32}\cos(\pi - \theta_2) \\ l_4^2 &= a_{41}^2 + a_{42}^2 - 2a_{41}a_{42}\cos(\theta_2) \end{aligned} \tag{25}$$

We constructed a right triangle to calculate l_5 and l_6

$$\begin{aligned} l_5^2 &= (a_{51} + a_{01})^2 + (a_{52} + a_{02})^2 - 2(a_{51} + a_{01})(a_{52} + a_{02}) \\ &\quad \times \cos(\pi - \theta_1 - \theta_2) \\ l_6^2 &= (a_{01} - a_{61})^2 + (a_{02} - a_{62})^2 - 2(a_{01} - a_{61})(a_{02} - a_{62}) \\ &\quad \times \cos(\pi - \theta_1 - \theta_2), \end{aligned} \tag{26}$$

where a_{01} and a_{02} can be obtained by Sines Law

$$a_{01} = \frac{d_1 \sin(\theta_2)}{\sin(\theta_1 + \theta_2)}, \quad a_{02} = \frac{d_1 \sin(\theta_1)}{\sin(\theta_1 + \theta_2)} \tag{27}$$

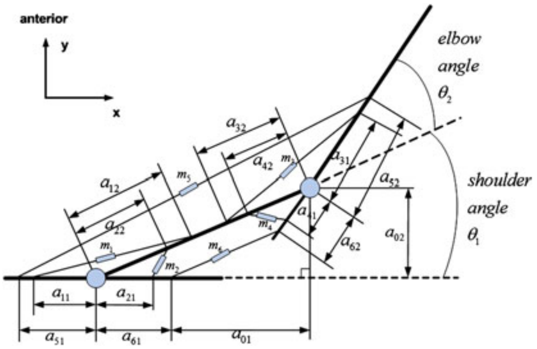

Fig. 2 Coordinate system. The attached positions of the muscles are defined in the coordinate system

After simplification, we finally get

$$
\begin{aligned}
l_1 &= \left(a_{11}^2 + a_{12}^2 + 2a_{11}a_{12}\cos(\theta_1)\right)^{1/2} \\
l_2 &= \left(a_{21}^2 + a_{22}^2 - 2a_{21}a_{22}\cos(\theta_1)\right)^{1/2} \\
l_3 &= \left(a_{31}^2 + a_{32}^2 + 2a_{31}a_{32}\cos(\theta_2)\right)^{1/2} \\
l_4 &= \left(a_{41}^2 + a_{42}^2 - 2a_{41}a_{42}\cos(\theta_2)\right)^{1/2} \\
l_5 &= \Big((a_{51}+a_{01})^2 + (a_{52}+a_{02})^2 + 2(a_{51}+a_{01})(a_{52}+a_{02}) \\
&\qquad \times \cos(\theta_1+\theta_2)\Big)^{1/2} \\
l_6 &= \Big((a_{01}-a_{61})^2 + (a_{02}-a_{62})^2 + 2(a_{01}-a_{61})(a_{02}-a_{62}) \\
&\qquad \times \cos(\theta_1+\theta_2)\Big)^{1/2}
\end{aligned}
\tag{28}
$$

Assuming the kinematics between the length of muscle and the angle of joint is given as follows

$$L = Q(\Theta) \tag{29}$$

where

$$
\begin{aligned}
L &= [\,l_1 \quad l_2 \quad l_3 \quad l_4 \quad l_5 \quad l_6\,]^{\mathrm{T}} \\
\Theta &= [\,\theta_1 \quad \theta_2\,]^{\mathrm{T}}
\end{aligned}
\tag{30}
$$

then the derivative of Eq. (29) is

$$\dot{L} = J_{\mathrm{m}}\dot{\Theta} \tag{31}$$

where J_m is a Jacobian matrix between the joint space and the muscle space. It has a format as

$$
J_m = \begin{bmatrix}
J_{m,1,1} & J_{m,1,2} \\
J_{m,2,1} & J_{m,2,2} \\
J_{m,3,1} & J_{m,3,2} \\
J_{m,4,1} & J_{m,4,2} \\
J_{m,5,1} & J_{m,5,2} \\
J_{m,6,1} & J_{m,6,2}
\end{bmatrix},
\tag{32}
$$

where

$$
\begin{aligned}
J_{m,1,1} &= \frac{-a_{11}a_{12}\sin(\theta_1)}{\sqrt{a_{11}^2 + 2\cos(\theta_1)a_{11}a_{12} + a_{12}^2}} \\
J_{m,2,1} &= \frac{-a_{21}a_{22}\sin(\theta_1)}{\sqrt{a_{21}^2 + 2\cos(\theta_1)a_{21}a_{22} + a_{22}^2}} \\
J_{m,3,2} &= \frac{-a_{31}a_{32}\sin(\theta_2)}{\sqrt{a_{31}^2 + 2\cos(\theta_2)a_{31}a_{32} + a_{32}^2}} \\
J_{m,4,2} &= \frac{a_{41}a_{42}\sin(\theta_2)}{\sqrt{a_{41}^2 + 2\cos(\theta_2)a_{41}a_{42} + a_{42}^2}} \\
J_{m,1,2} &= J_{m,2,2} = J_{m,3,1} = J_{m,4,1} = 0
\end{aligned}
\tag{33}
$$

$J_{m,5,1}$, $J_{m,5,2}$, $J_{m,6,1}$ and $J_{m,6,2}$ are long equations and we do not provide them here. The derivation of the above Jacobian matrix can be done by Matlab Symbolic Toolbox.

4.2 Muscle force distribution

The relationship between muscle forces and joint torques can be derived by the principle of virtual work as

$$\tau = f(F_m) = J_m^{\mathrm{T}} F_m. \tag{34}$$

Hence, the inverse relation between the joint torques and the muscle forces can be expressed as

$$\tau_{\mathrm{inv}} = f^{-1}(\tau) = \left(J_m^{\mathrm{T}}\right)^{+}\tau \tag{35}$$

where

$$\left(J_m^{\mathrm{T}}\right)^{+} = J_m\left(J_m^{\mathrm{T}} J_m\right)^{-1} \tag{36}$$

is pseudo-inverse matrix of J_m^{T}. The above muscle force distribution solution satisfies

$$
\begin{aligned}
&\min\|F_m\| \\
&s.t.\ J_m^{\mathrm{T}} F_m = \tau
\end{aligned}
\tag{37}
$$

which means the pseudoinverse is an optimization solution to obtain minimums muscle distribution force. However, the above solution does not consider physical constraints, such as the maximum output force of muscle is limited, muscles can only contract, etc. To involve these constraints, we define F_{in} as a voluntary vector having the same dimension with F_m which expresses the internal forces generated by redundant muscles. Then, we can define the internal force in F_m space, i.e.

$$g(F_{\mathrm{in}}) = \left(I - \left(J_m^{\mathrm{T}}\right)^{+} J_m^{\mathrm{T}}\right) F_{\mathrm{in}} \tag{38}$$

where I is an identity matrix having the same dimension with muscle space. According to Moore–Penrose pseudo-inverse, $g(F_{\mathrm{in}})$ is orthogonal with the pseudo-inverse solution. Thus, we can choose any vector as F_{in}. Below, we give a gradient direction for F_{in} to make F_m satisfy boundary constraints.

Here, we assume that each muscle force is limited in the interval from $F_{m,i,\min}$ to $F_{m,i,\max}$ for $1 \le i \le 6$. Our objective is to choose a gradient direction to make each element of $F_{m,i}$ $(1 \le i \le 6)$ equal or greater than $F_{m,i,\min}$, and equal or

Table 1 Anthropological parameter values

Segment	Upper arm	Lower arm
Length (m)	0.282	0.269
Mass (kg)	1.980	1.180
MCS Pos (m)	0.163	0.123
I_{11} (kg m^2)	0.013	0.007
I_{22} (kg m^2)	0.004	0.001
I_{33} (kg m^2)	0.011	0.006

The parameter setting of the arm is based on a real human data [40]

MCS Pos position of the mass center

less than $F_{m,i,\max}$. Considering the muscle fatigue, one reasonable way is to make each output force of the muscles to be around at the middle magnitude between $F_{m,i,\min}$ and $F_{m,i,\max}$. The physical meaning of this method is to distribute load to all the muscles in their proper load interval, so that they can continue working for a longer time. Based on these considerations, we choose a function h as

$$h(F_m) = \sum_{j=1}^{6} \left(\frac{\tau_{\mathrm{inv},i} - F_{m,i,\mathrm{mid}}}{F_{m,i,\mathrm{mid}} - F_{m,i,\max}} \right)^2 \tag{39}$$

where

$$\begin{aligned} &0 \le F_{m,i,\min} \le \tau_{\mathrm{inv},i} \le F_{m,i,\max} \\ &F_{m,i,\mathrm{mid}} = \frac{F_{m,i,\min} + F_{m,i,\max}}{2} \\ &\quad i = 1, 2, \ldots 6 \end{aligned} \tag{40}$$

then we chose F_{in} as the gradient of the function h, i.e.

$$F_{\mathrm{in}} = K_{\mathrm{in}} \frac{\partial h(\tau_{\mathrm{inv}})}{\partial \tau_{\mathrm{inv}}} = K_{\mathrm{in}} \nabla_h = K_{\mathrm{in}} \begin{bmatrix} 2 \cdot \frac{\tau_{\mathrm{inv},1} - F_{m,1,\mathrm{mid}}}{F_{m,1,\mathrm{mid}} - F_{m,1,\max}} \\ 2 \cdot \frac{\tau_{\mathrm{inv},2} - F_{m,2,\mathrm{mid}}}{F_{m,2,\mathrm{mid}} - F_{m,2,\max}} \\ 2 \cdot \frac{\tau_{\mathrm{inv},3} - F_{m,3,\mathrm{mid}}}{F_{m,3,\mathrm{mid}} - F_{m,3,\max}} \\ 2 \cdot \frac{\tau_{\mathrm{inv},4} - F_{m,4,\mathrm{mid}}}{F_{m,4,\mathrm{mid}} - F_{m,4,\max}} \\ 2 \cdot \frac{\tau_{\mathrm{inv},5} - F_{m,5,\mathrm{mid}}}{F_{m,5,\mathrm{mid}} - F_{m,5,\max}} \\ 2 \cdot \frac{\tau_{\mathrm{inv},6} - F_{m,6,\mathrm{mid}}}{F_{m,6,\mathrm{mid}} - F_{m,6,\max}} \end{bmatrix} \tag{41}$$

where K_{in} is a scalar matrix. It is very easy to prove that the direction of F_{in} points to $F_{m,i,\mathrm{mid}}$. According to the computation in Eqs. (35) and (41), the muscle force is calculated as

$$F_m = \tau_{\mathrm{inv}} + g(F_{\mathrm{in}}) \tag{42}$$

The procedures to compute muscle force are concluded as the following algorithm.

Algorithm 1 Muscle force distribution

```
• Step 1  Computing Jacobian matrix $J_m$ (Eq. (32)) according to Eq. (33).
• Step 2  Computing pseudoinverse matrix $\left(J_m^T\right)^+$ according to Eq. (36).
• Step 3  Computing $\tau_{inv}$ according to Eq. (35).
• Step 4  Computing internal force $F_{in}$ according to Eq. (41). Here, $\tau_{inv,i}$ $(1 \le i \le 6)$ comes
          from the result in Step 3.
• Step 5  Computing $g(F_{in})$ according to Eq. (38).
• Step 6  Computing muscle force by Eq. (42) where $\tau_{inv}$ comes from Step 3 and $g(F_{in})$
          comes from Step 5.
```

5 Bending-streching movement simulation

The performance of the proposed muscle force computation method was tested by simulation. The desired movement is bending the upper arm and lower arm from 0 rad to $\pi/2$ rad and then stretching them back to 0 rad. The total simulation time was 10 s.

5.1 Arm model parameter setting

The parameters of the robot arm are based on the real data of a human upper limb. The setting of length, mass, mass center position and inertia coefficients is shown in Table 1. The anthropological data come from [40]. Without loss of generality, the muscle configuration coefficients (in Eq. 28) are set as $a_{ij} = 0.1\,\mathrm{m}$ $(1 \le i \le 6, 1 \le j \le 2)$.

5.2 Computational coefficient setting

There are three groups of parameters need to set, including the parameters for sliding mode control, the parameters for parameter adaptation, and the parameters for muscle force computation. These parameters are set as follows.

In this research, the control parameters are set (Eq. 12) as

$$K = 20 \cdot \mathrm{Diag}([1 \quad 1]) \tag{43}$$

the adaptation parameters (Eq. 24) are set as

$$\begin{aligned} \Gamma^{-1} &= 0.0015 \cdot \mathrm{Diag}([1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1]) \\ 2\Xi &= 0.001 \cdot \mathrm{Diag}([1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1]) \end{aligned} \tag{44}$$

and the muscle force computation parameters (Eq. 41) are set as

$$K_{\mathrm{in}} = 200 \cdot \mathrm{Diag}([1 \quad 3 \quad 1 \quad 1 \quad 1 \quad 2]) \tag{45}$$

$$F_{m,i,\min} = 0, \quad F_{m,i,\max} = 1000 \ (1 \le i \le 6), \tag{46}$$

where $\mathrm{Diag}(\cdot)$ is a diagonal matrix with diagonal elements being as $(\cdot)$.

5.3 Control performance

According to the bending-stretching movement, two sinusoidal waves are set as reference signals for q_1 and

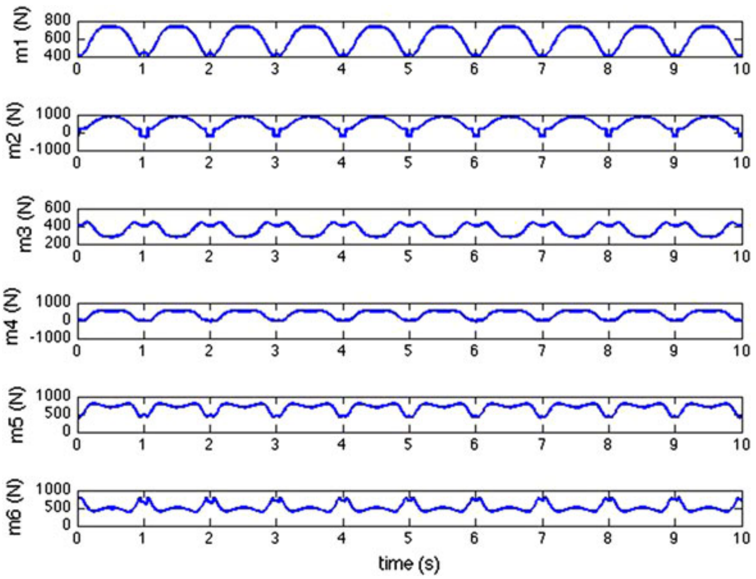

Fig. 3 Muscle forces. Six muscle forces are computed to drive the arm to track the desired motion

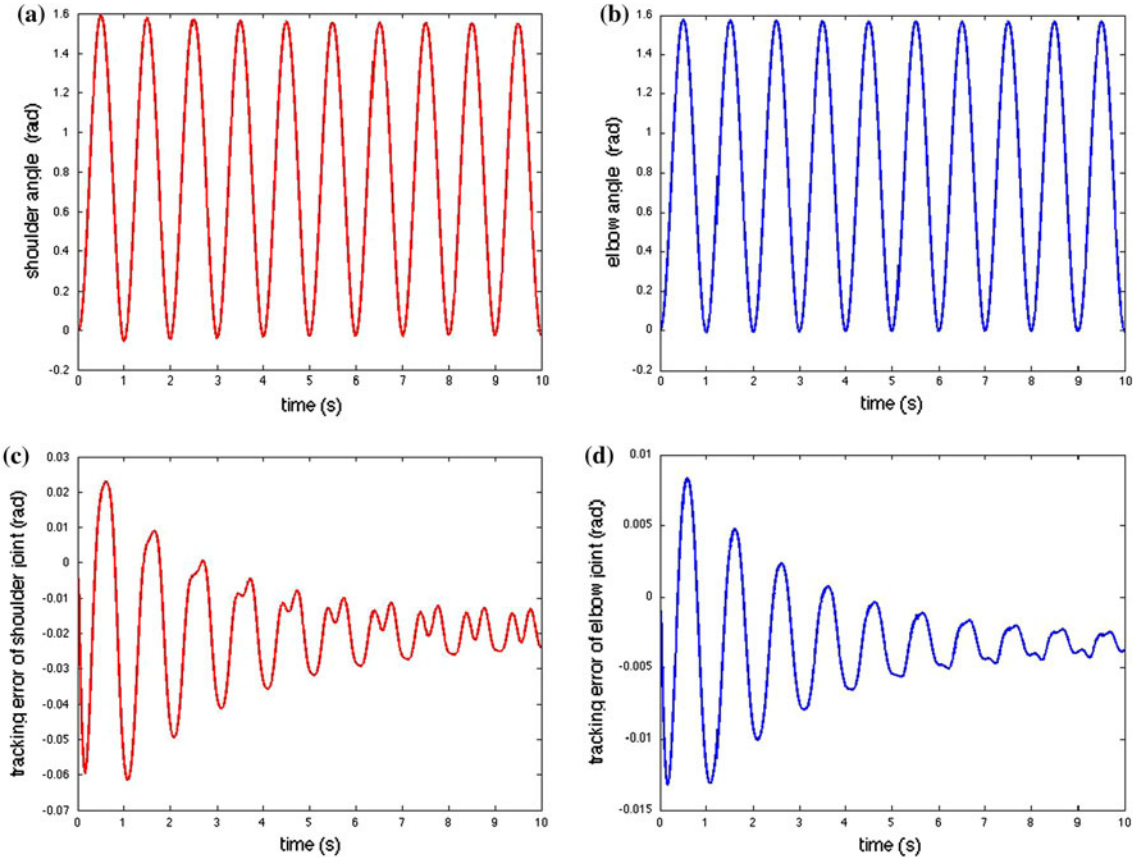

Fig. 4 Arm movement performance. The proposed muscle control method can drive the arm to track the desired motion accurately. **a** Shoulder angle. **b** Elbow angle. **c** Tracking error of the shoulder joint. **d** Tracking error of the elbow joint

q_2. The frequency of the two waves is set as 2π. Initial states of q_1 and q_2 are set as zero. Based on the joint torque coming from sliding mode control, we compute the muscle force according to Algorithm 1. The computed 6 muscle forces are shown in Fig. 3. All the muscle forces are in the range of $\left[F_{m,i,\min}, F_{m,i,\max}\right]$ for $(1 \leq i \leq 6)$. These muscle forces are optimized to be around $F_{m,i,\text{mid}}$.

These computed muscle forces are used to control the humanoid robot arm model. The shoulder angle and elbow angle are shown in Fig. 4a, b, respectively. Compared with the desired trajectory, the tracking error of shoulder joint and elbow joint is shown in Fig. 4c, d, respectively. It is clear that the tracking performance is good. Additionally, both the two tracking errors decrease gradually. The reason is that the parameter update makes the estimated model parameters to approach the real ones gradually.

5.4 Parameter adaptation

In this research, in order to test the functionality of the designed parameter adaptation method, we set the initial estimated model parameters $\hat{H}$, $\hat{C}$ and $\hat{G}$ to be zero matrix (or zero vector) at the beginning. After that the parameter adaptation method adjusts these parameters based on the tracking error and the prediction error. Figure 5 shows the parameters H, $\hat{H}$ and C, $\hat{C}$ in the time interval $[0, 1]$s. We took four snapshots of these parameters at the moment 0, 1/3, 2/3, 1 s, respectively. It is noted that the estimated parameters do not coincide with real parameters, i.e., because the dynamic feature of the model is partly stimulated. The more complicated the movement, the more consistent is between the estimated parameters and real parameters.

5.5 Animation

A humanoid robot arm is visualized by Simulink (SimMechanics Toolbox). The arm model consists of three parts: torso, right humerus, and right ulna radius hand. The

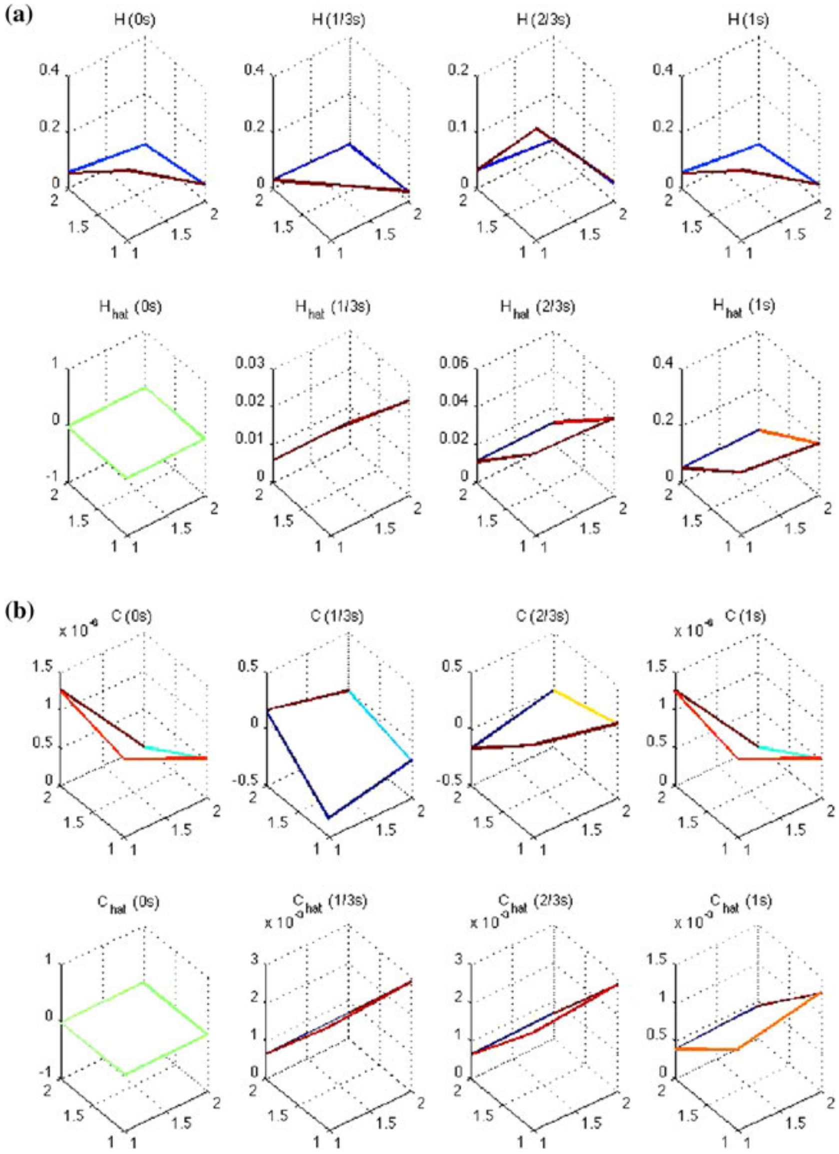

Fig. 5 Parameter update. Four snapshots of H, $\hat{H}$, C, and $\hat{C}$ are taken averagely from 0 to 1 s. The reason for the difference of the estimated parameter and the real ones is that the system dynamics has not been fully stimulated. But this inconsistency does not influence the control performance. **a** Snapshots of H and $\hat{H}$. **b** Snapshots of C and $\hat{C}$

three parts are created by 6 bones, 1 bone, and 29 bones, respectively. The polygon files of these bones come from SIMM. To make Simulink be able to import these polygons, we converted the format of polygon files from.vtp file to.stl file. Figure 6 shows three phases (i.e., start phase, middle phase, and end phase) of the arm gesture change in a bending-stretching movement cycle.

6 Conclusion

After considering the numerous advantages of bi-articular muscles, such as robustness and potential for load distribution, in this paper we addressed the analogous problem applied not to humans, but to robots; namely, the problem of motor control of over-actuated robot arms. In our proposed solution, a relatively mature control method (i.e. sliding mode control) was chosen to get joint torque first, and then the joint torque was distributed to the muscle forces. The muscle force was computed based on a Jacobian matrix between joint torque space and muscle force space. In addition, internal forces were used to optimize the computed muscle forces towards two goals: not only to make sure that each muscle force is in its force boundary, but also to make the muscles work in the middle of their working range, which is considered best in terms of fatigue. Besides, all the dynamic parameters were updated in real-time. A method was also proposed to use prediction error to accelerate the convergence speed of parameter. Apart from designing our over-actuated arm motor control method, we empirically evaluated it for the case of bending-stretching movements. The results clearly illustrate the effectiveness of our method towards achieving the desired kinetic as well as load distribution characteristics.

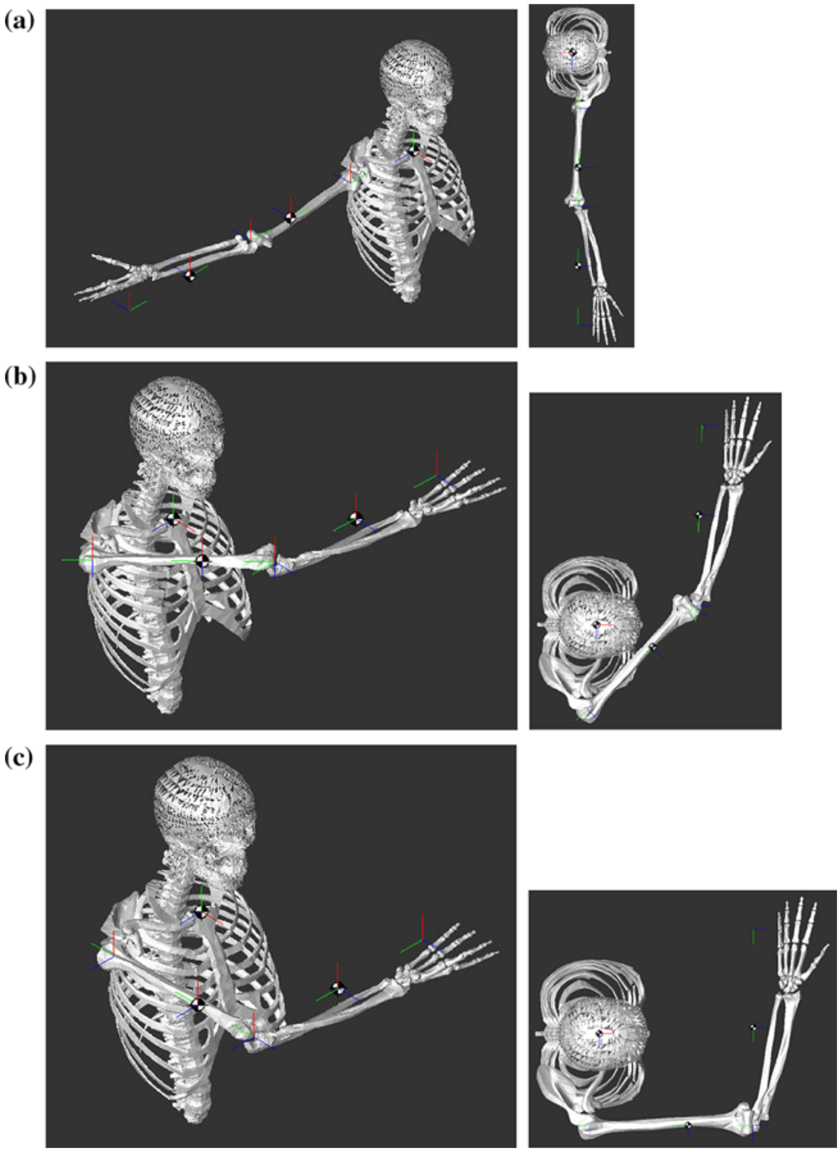

Fig. 6 Arm movement in the isometric and top view. The arm bends and stretches repeatedly. The animation is made by Simulink SimMechanics Toolbox. This visual model is consisted of three parts: torso (6 bones), right_humerus (1 bone) and right_ulna_radius_hand (29 bones). **a** Start phase. **b** Middle phase. **c** End phase

References

1. Oh S, Hori Y (2009) Development of two-degree-of-freedom control for robot manipulator with biarticular muscle torque. In: American Control Conference 2009, pp 325–330
2. Klug S, Mohl B, Stryk OV, Barth O (2005) Design and application of a 3 DOF bionic robot arm. In: 3rd International symposium on adaptive motion in animals and machines 2005, pp 1–6
3. Potkonjak V, Jovanovic KM, Milosavljevic P, Bascarevic N, Holland O (2011) The puller-follower control concept in the multi-jointed robot body with antagonistically coupled compliant drives. In: IASTED international conference on robotics 2011, pp 375–381
4. Tahara K, Luo Z, Arimoto S, Kino H (2005) Sensor-motor control mechanism for reaching movements of a redundant musculo-skeletal arm. J Robot Syst 22:639–651
5. Ivancevic VG, Ivancevic TT (2005) Natural biodynamics. World Scientific, Singapore
6. Crago PE (2000) Creating neuromusculoskeletal models. Biomechanics and neural control of posture and movement. Springer, Berlin, pp 119–133
7. Winter DA (1990) Biomechanics and motor control of human movement. Wiley, New York
8. Tzafestas S, Raibert M, Tzafestas C (1996) Robust sliding-mode control applied to a 5-link biped robot
9. Kumamoto M (2006) Revolution in humanoid robotics: evolution of motion control. Tokyo Denki University Publisher, Tokyo
10. Liegeois A (1977) Automatic supervisory control of the configuration and behavior of multibody mechanisms. IEEE Trans Syst Man Cybern 7:868–871
11. Manal K, Gonzalez RV, Lloyd DG, Buchanan TS (2002) A real-time EMG-driven virtual arm. Comput Biol Med 32:25–36
12. Pandy MG (2001) Computer modeling and simulation of human movement. Annu Rev Biomed Eng 3:245–273
13. Tee KP, Burdet E, Chew CM, Milner TE (2004) A model of force and impedance in human arm movements. Biol Cybern 90:368–375
14. Hogan N (1984) Adaptive control of mechanical impedance by coactivation of antagonist muscles. IEEE Trans Autom Control 29:681–690
15. Anderson FC, Pandy MG (2001) Dynamic optimization of human walking. J Biomech Eng 125:381–390
16. Neptune RR (1999) Optimization algorithm performance in determining optimal controls in human movement analyses. J Biomech Eng 121:249–252
17. Crowninshield RD, Brand RA (1981) A physiologically based criterion of muscle force prediction in locomotion. J Biomech 14:793–801
18. Glitsch U, Baumann W (1997) The three-dimensional determination of internal loads in the lower extremity. J Biomech 30:1123–1131
19. Hogan N (1984) An organizing principle for a class of voluntary movements. J Neurosci 4:2745–2754
20. Patriarco AG, Mann RW, Simon SR, Mansour JM (1981) An evaluation of the approaches of optimization models in the prediction of muscle forces during human gait. J Biomech 14:513–525
21. Anderson FC, Pandy MG (1999) A dynamic optimization solution for vertical jumping in three dimensions. Comput Methods Biomech Biomed Eng 2:201–231
22. Damsgaard M, Rasmussen J, Christensen ST, Surma E, Zee MD (2006) Analysis of musculoskeletal systems in the Anybody modeling system. Simul Model Pract Theory 14:1100–1111
23. Delp SL, Anderson FC, Arnold AS, Loan P, Habib A, John CT et al (2007) Opensim: open-source software to create and analyze dynamic simulations of movement. IEEE Trans Biomed Eng 54:1940–1950
24. Kawato M (1999) Internal models for motor control and trajectory planning. Curr Opin Neurobiol 9:718–727
25. Gottlieb GL (1993) A computational model of the simplest motor program. J Mot Behav 25:153–161
26. Karniel A, Inbar GF (1996) A model for learning human reaching-movements. Biol Cybern 77:173–183
27. Katayama M, Kawato M (1993) Virtual trajectory and stiffness ellipse during multijoint arm movement predicted by neural inverse models. Biol Cybern 69:353–362
28. Chaumette F, Hutchinson S (2006) Visual servo control—part I: basic approaches. IEEE Robot Autom Mag 13:82–90
29. Yoshikawa T (1984) Analysis and control of robot manipulators with redundancy. In: Robotics research: the first international symposium. MIT Press, Cambridge, pp 735–748
30. Maciejewski AA, Klein CA (1985) Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments. Int J Robot Res 4:109–117
31. Hollerbach JM, Suh KC (1987) Redundancy resolution of manipulators through torque optimization. IEEE J Robot Autom 3:308–316
32. Khosla PK, Kanade T (1988) Experimental evaluation of nonlinear feedback and feedforward control schemes for manipulators. Int J Robot Res 7:18–28
33. Heim A, Stryk OV (2000) Trajectory optimization of industrial robots with application to computer-aided robotics and robot controllers. Optimization 47:407–420
34. Graupe D (1972) Identification of systems. Litton Educational Publishing, New York
35. Zhou K, Doyle JC (1997) Essentials of robust control. Prentice Hall, New Jersey
36. Slotine JE, Li W (1991) Applied nonlinar control. Prentice Hall, New Jersey
37. Brown M, Harris C (1994) Neurofuzzy adaptive modelling and control. Prentice Hall, New Jersey
38. Slotine JE, Li W (1988) Adaptive manipulator control: a case study. IEEE Trans Autom Control 33:995–1003
39. Dong H, Luo Z, Nagano A (2010) Adaptive attitude control for redundant time-varying complex model of human body in the nursing activity. J Robot Mechatron 22:418–429
40. Nagano A, Yoshioka S, Komura T, Himeno R, Fukashiro S (2005) A three-dimensional linked segment model of the whole human body. Int J Sport Health Sci 3:311–325