



From Rough to Precise: Human-Inspired Phased Target Learning Framework for Redundant Musculoskeletal Systems

Junjie Zhou^{1,2,3}, Jiahao Chen^{2,3,4}, Hu Deng^{1,3} and Hong Qiao^{1,2,5*}

¹ State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China, ² School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China, ³ Beijing Key Laboratory of Research and Application for Robotic Intelligence of “Hand–Eye–Brain” Interaction, Beijing, China,

⁴ Research Center for Brain-Inspired Intelligence, Institute of Automation, Chinese Academy of Sciences, Beijing, China,

⁵ CAS Center for Excellence in Brain Science and Intelligence Technology, Shanghai, China

Redundant muscles in human-like musculoskeletal robots provide additional dimensions to the solution space. Consequently, the computation of muscle excitations remains an open question. Conventional methods like dynamic optimization and reinforcement learning usually have high computational costs or unstable learning processes when applied to a complex musculoskeletal system. Inspired by human learning, we propose a phased target learning framework that provides different targets to learners at varying levels, to guide their training process and to avoid local optima. By introducing an extra layer of neurons reflecting a preference, we improve the Q-network method to generate continuous excitations. In addition, based on information transmission in the human nervous system, two kinds of biological noise sources are introduced into our framework to enhance exploration over the solution space. Tracking experiments based on a simplified musculoskeletal arm model indicate that under guidance of phased targets, the proposed framework prevents divergence of excitations, thus stabilizing training. Moreover, the enhanced exploration of solutions results in smaller motion errors. The phased target learning framework can be expanded for general-purpose reinforcement learning, and it provides a preliminary interpretation for modeling the mechanisms of human motion learning.

Keywords: musculoskeletal system, human-inspired motion learning, noise in nervous system, reinforcement learning, phased target learning

OPEN ACCESS

Edited by:

Changhong Fu,
Tongji University, China

Reviewed by:

Chenguang Yang,
University of the West of England,
United Kingdom
Zhihao Xu,
Guangdong Institute of Intelligent
Manufacturing, Guangdong Academy
of Sciences, China

*Correspondence:

Hong Qiao
hong.qiao@ia.ac.cn

Received: 12 April 2019

Accepted: 15 July 2019

Published: 31 July 2019

Citation:

Zhou J, Chen J, Deng H and Qiao H
(2019) From Rough to Precise:
Human-Inspired Phased Target
Learning Framework for Redundant
Musculoskeletal Systems.
Front. Neurobot. 13:61.
doi: 10.3389/fnbot.2019.00061

1. INTRODUCTION

Research on human-like musculoskeletal robots has become multidisciplinary in recent years, as it involves fields such as neuroscience and materials science for modeling and implementing musculoskeletal motor systems. In fact, this branch of robotics mainly comprises muscle models (actuators), skeletal systems (supporting structure), and methods for motion control and learning (control systems). Related work can roughly be divided into two types, namely, muscle dynamics modeling along with hardware design (Jäntschi et al., 2013; Kurumaya et al., 2016; Asano et al., 2017) and musculoskeletal robot control (Pennestrì et al., 2007; Jagodnik and van den Bogert, 2010; Tahara and Kino, 2010). Although most studies have been focused on the first type, the development of neuroscience has gradually increased the research on human-inspired control.

As a multibody mechanical system (Stoianovici and Hurmuzlu, 1996; Shi and McPhee, 2000) comprising muscles and joints, the human musculoskeletal system has several advantages. For instance, muscle redundancy maintains the reliable operation of the musculoskeletal system when some muscles are fatigued or even damaged. Under control of the central nervous system, the musculoskeletal system can accomplish accurate and fine manipulation (Rasmussen et al., 2001; Chen et al., 2018). To unveil the mechanisms that provide such advantages, Hill studied the contraction properties of muscles, establishing the Hill model (Hill, 1938). From this fundamental work, a series of muscle dynamic models have been proposed (Huxley and Niedergerke, 1954; Eisenberg et al., 1980; Zahalak and Ma, 1990), but all of them present specific limitations. For instance, the simple second-order model (Cook and Stark, 1968; Agarwal et al., 1970) lacks independent nodal locations for external input signals, which indirectly affect the output. The Huxley contraction model (Huxley, 1957) is highly complex and no general-purpose method has been developed to obtain its parameters (Winters and Stark, 1987). The Hill model presents difficulties in measuring the fiber length during motion (Arnold and Delp, 2011).

Research has also been devoted to design hardware for emulating muscle characteristics. The Anthrob muscle unit (Jäntschi et al., 2013) and the sensor–driver integrated muscle module (Asano et al., 2015) try to resemble muscular structures. However, the weight and size of motors make hardware models notably diverge from biological muscles. Furthermore, resembling tiny human muscles through hardware design is difficult, thus undermining their applicability. In materials science, the synthesis of ideal materials for artificial muscles is being pursued to achieve the characteristics of biological muscles regarding size, weight, stiffness, and dynamic behavior. New materials for artificial muscles usually share some problems, including unsafe voltages and low strain. Accessory equipment can partly adjust the characteristics of materials. For instance, liquid-vapor transition has been used on a soft composite material (Miriyeve et al., 2017) for implementation as an actuator in a variety of robotic applications. In addition, a coiled polymer muscle (Haines et al., 2014) controlled by varying water temperature prevents dependence on electricity. Hence, advanced design methods and materials seem promising to develop artificial muscles that closely reflect the dynamics of their biological counterparts.

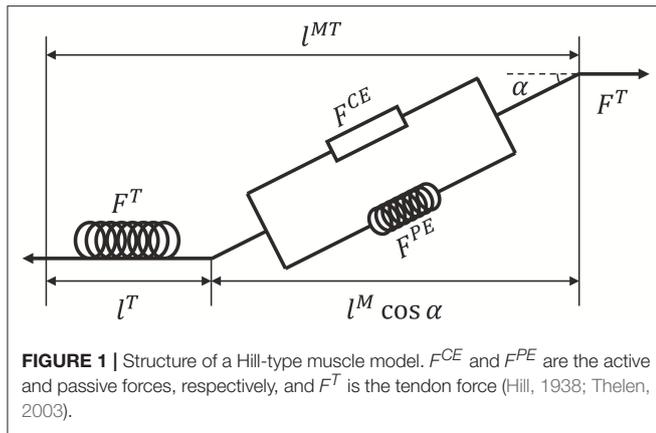
Based on the abovementioned models, control systems developed for musculoskeletal robotics also face challenges. Redundant muscles and extremely complex tendon forces impose several barriers for direct solutions of muscle excitation. Widely used methods, such as inverse dynamics with static optimization (Crowninshield and Brand, 1981), computed muscle control (Thelen et al., 2003), proportional-derivative control (Jagodnik and van den Bogert, 2010), and PI-type iterative learning control (Tahara and Kino, 2010), are used to regulate musculoskeletal systems. Although some conventional methods, such as computed muscle control, theoretically compute muscle excitation signals, they also demand intensive computations for sophisticated processes (Chen et al., 2018).

In addition, these control strategies are hardly supported by biological evidence showing that they resemble the approach of human motion learning.

In recent years, reinforcement learning has become a popular control method in robotics as it provides a natural-like approach to learn from the environment. In fact, as a method that fosters interaction with uncertain environments, reinforcement learning allows a learner to observe the environment and then execute appropriate actions. The environment provides rewards for each action, and the learner aims to maximize its rewards during decision-making. This learning process is similar to that of humans and animals (Sutton and Barto, 2018). Studies in neuroscience (Schultz et al., 1997; Law and Gold, 2009) verify this principle, and hence it is reasonable to consider human-like learning from the viewpoint of reinforcement learning (Tesauro, 1995; Diuk et al., 2008; Riedmiller et al., 2009). Deep neural networks are adopted to implement reinforcement learning. Specifically, the deep Q-network (Mnih et al., 2015) uses a deep convolutional neural network to estimate the action-value function, making deep reinforcement learning a powerful weapon for a myriad of applications (Van Hasselt et al., 2016; Wang et al., 2016; Hou et al., 2017). However, when applied to the musculoskeletal system, the performances of deep neural networks can be unstable. Given muscle redundancy in the musculoskeletal system, the additional dimensions expand the solution space, hindering optimization through reinforcement learning.

In this study, we focused on the unstable training of musculoskeletal systems and the expanded solution space of excitations to provide three contributions. (1) The learning goal of humans, changes stepwise as learning proceeds over advancing levels. For example, running requires higher physical coordination than walking, and one cannot run before learning to walk. Thereby, the learner target evolves from walking to running during this process. Based on this principle, we propose the phased target learning (PTL) framework that reduces the computational cost for exploration in a high-dimensional solution space. In addition, phased targets guide the convergence of excitations to the expected value during training. (2) As sensory information may be encoded by opposite tuning neurons (Romo and de Lafuente, 2013), we improve an MLP-based Q-network by introducing an extra layer of neurons reflecting preference and using various relative action probabilities from value functions for obtaining continuous outputs to control a musculoskeletal arm model. (3) As noise exists in the nervous system (Aldo et al., 2008) and based on information transmission in the human nervous system (Dhawale et al., 2017), we introduce two noise sources at the sensor and execution levels into the proposed PTL framework. These noise sources increase the exploration capacity in the solution space during training and strengthen the control robustness.

In this paper, in section 2, we introduce the muscle dynamics, the structure of the arm model, and detail the musculoskeletal system considered in this study. Moreover, optimization of the proposed PTL framework is outlined. Then, the PTL framework with the biological noise sources is introduced in section 3.



Experimental results and conclusions are presented in sections 4 and 5, respectively.

2. MUSCULOTENDON MODEL AND MUSCULOSKELETAL ARM MODEL

Modeling muscles is difficult because most parameters cannot be measured precisely in real time (Arnold and Delp, 2011). According to the Hill model (Hill, 1938), which defines that a muscle is made up of separate elements, such as contractile elements (CE), passive elements (PE), and series elastic elements (SEE) (Zajac, 1989; Thelen et al., 2003), we design a control framework for musculoskeletal systems.

2.1. Musculotendon Model

To determine the way a human can control complex muscle systems, a muscle dynamic model is necessary. Let $u \in [0, 1]$ denote an idealized muscle excitation signal. According to a nonlinear first-order differential Equation (1), muscle activation signal a can be computed (Thelen, 2003):

$$\frac{da}{dt} = \frac{u - \hat{a}}{\tau(u, a)}, \quad (1)$$

where τ varies according to idealized muscle excitation signal u and activation signal a (Winters, 1995), \hat{a} is the activation signal after normalization, and a is transmitted to the muscle contraction dynamic model as a final control signal.

Before introducing the muscle contraction dynamics, the structure of a Hill-type muscle model is shown in **Figure 1**, where l^T and l^M are the lengths of the tendon and muscle fiber, respectively, and α is the muscle pennation angle (Garner and Pandy, 2003). When the activation signal a is transmitted to the muscle, the corresponding muscle force is generated by contraction. Then, the muscle force pulls the skeletons to generate motion or to maintain the balance of forces.

Suppose that signal u is known. To calculate tendon force F^T , some assumptions are required. First, $F^T, F^{CE}, F^{PE} > 0$ because muscles move the skeleton by tension instead of thrusting. Second, the change of muscle width can be ignored during muscle

contraction (Matthew et al., 2013). Third, muscle mass can be ignored. Using these assumptions, the dynamics of muscles can be described. Specifically, a pennation angle α can be obtained from

$$l_s^M \sin(\alpha_0) = l^M(t) \sin(\alpha(t)), \quad (2)$$

where l_s^M and α_0 are the slack length of a muscle fiber and initial pennation angle, respectively, which also define the initial muscle width, $l^M(t)$ and $\alpha(t)$ are the length of the muscle fiber and pennation angle at time t , respectively. From $\alpha(t)$, tendon force F^T can be computed by a piecewise nonlinear equation (Proske and Morgan, 1987; Thelen, 2003). In addition, the contraction velocity of a muscle fiber is necessary for the model. To determine this velocity, active force F^{CE} produced by the contractile element should be obtained first. According to the geometric relationship between tendon and muscle fiber (**Figure 1**), F^{CE} can be calculated indirectly as follows:

$$F^{CE} = \frac{F^T}{\cos(\alpha)} - F^{PE}, \quad (3)$$

where F^{PE} is the passive force of the muscle fiber. During simulations, the muscle length sometimes causes numerical problems that result in $F^{CE} < 0$, which clearly violate the first assumption about muscles. Therefore, a constraint should be added to avoid exceptional cases:

$$F^{CE} = \max\{F^{CE}, 0\}. \quad (4)$$

Then, contraction velocity v^M can be computed by another piecewise non-linear equation (Matthew et al., 2013):

$$v^M = f_v^{-1} \left(\frac{F^{CE}}{af_l(l^M)} \right), \quad (5)$$

where f_v is the force-velocity function, f_v^{-1} is its inverse function, and f_l is a Gaussian function with variable l^M (Winters, 1990). As a key variable in the muscle dynamics model, $v^M(t)$ affects $l^M(t+1)$ at every timestep. Variable l^M is the fiber length and l^{MT} is the muscle length, which comprises fiber and tendon. Length l^M can be calculated directly using v^M and F^T , whereas l^{MT} can be measured. Consequently, if signal $u(t)$ is known, the contraction states of the muscle and tendon force $F^T(t)$ can be computed.

2.2. Musculoskeletal Arm Model

In the remainder of this section, we first establish a simplified arm model to connect muscles and bones. Then, we analyze the kinematic relationship between the arm model and muscle model. Finally, a control framework is outlined using this relationship.

According to the Newton-Euler equation (Zixing, 2000; Hahn, 2013), we establish a two degree-of-freedom model (**Figure 2**) that consists of two segments and four muscles. Then, expected torque τ_n at the joints can be calculated as

$$\tau_n = \frac{\partial W}{\partial \theta} = M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + G(\theta), \quad (6)$$

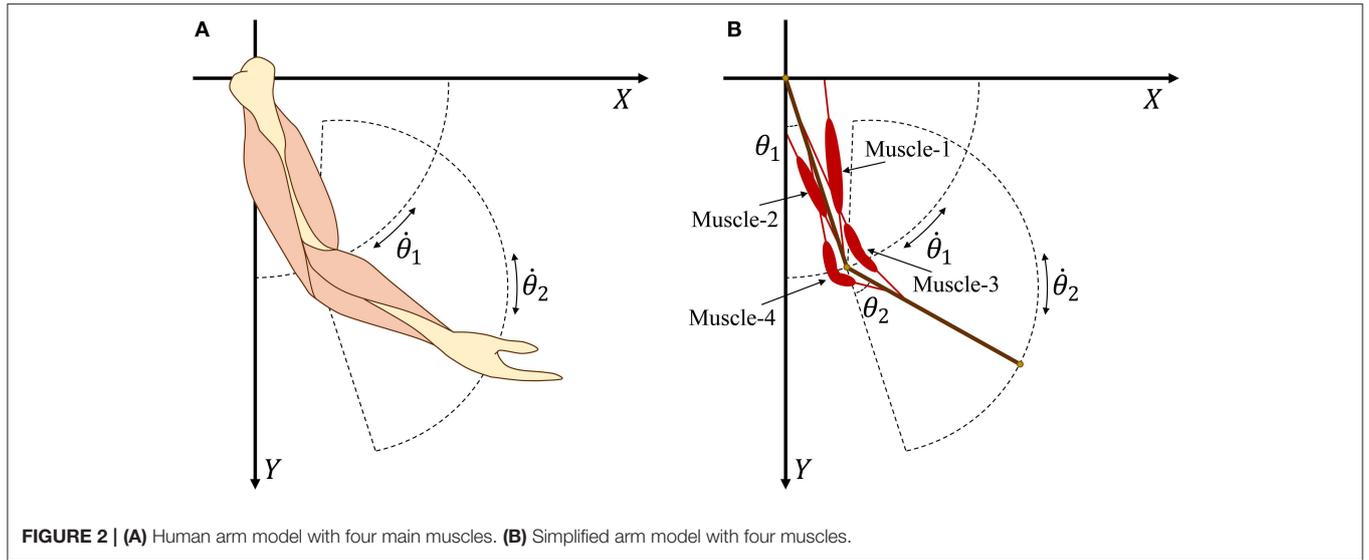


FIGURE 2 | (A) Human arm model with four main muscles. **(B)** Simplified arm model with four muscles.

where W is the work from external forces, $\dot{\theta}$ is the vector of rotational velocity, $\ddot{\theta}$ is the vector of rotational acceleration, $M(\theta) \in \mathbb{R}^{n \times n}$ and $C(\theta, \dot{\theta})\dot{\theta} \in \mathbb{R}^n$ is the inertia matrix and the centripetal and Coriolis force, respectively, and $G(\theta) \in \mathbb{R}^n$ is the gravitational force vector of our model. During forward calculation, Equation (6) provides a way to compute expected torques for known motion states. During inverse calculation, it can be used to compute actual angular acceleration.

2.3. Musculotendon Model Into Arm Model

In this section, we obtain the relationship between torques and motion states and define the adopted learning approach.

Unlike conventional robots that use a single joint motor to generate torque, each joint in a musculoskeletal system is usually affected by more than one muscle. Let τ_i be the muscle torque generated by muscle i :

$$\tau_i = F_i^T l_{i2} \sin \gamma_i, \quad i = 1, 2, \dots, n, \quad (7)$$

where F_i^T is the tendon force of muscle i and γ_i is the angle between the muscle and related bone. **Figure 3** provides geometric details of the muscles and bones. We set $m_1 = 2$ and $d_1 = 0.3$ as the mass and length of the upper arm, respectively, whereas $m_2 = 1.8$ and $d_2 = 0.3$ are the mass and length of the forearm, respectively. For the given geometry of the musculoskeletal model, the muscle torque can be written as

$$\begin{cases} \tau'_{n1} = \tau_1 - \tau_2 = F_1^T l_{12} \sin \gamma_1 - F_2^T l_{22} \sin \gamma_2 \\ \tau'_{n2} = \tau_3 - \tau_4 = F_3^T l_{32} \sin \gamma_3 - F_4^T l_{42} \sin \gamma_4 \end{cases} \quad (8)$$

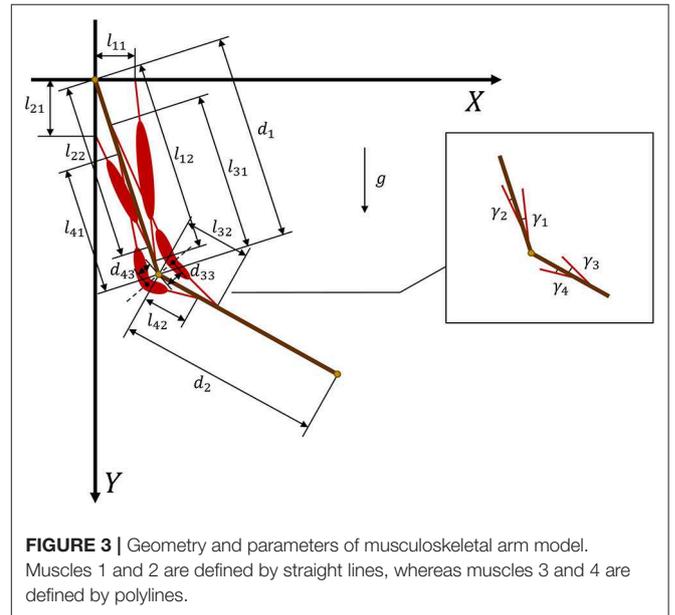


FIGURE 3 | Geometry and parameters of musculoskeletal arm model. Muscles 1 and 2 are defined by straight lines, whereas muscles 3 and 4 are defined by polylines.

In addition, the geometric parameters can be used to compute $\sin \gamma_i$:

$$\begin{cases} \sin \gamma_1 = \frac{l_{11} \cos \theta_1}{\sqrt{l_{11}^2 + l_{12}^2 - 2l_{11}l_{12} \sin \theta_1}} \\ \sin \gamma_2 = \frac{l_{21} \sin \theta_1}{\sqrt{l_{21}^2 + l_{22}^2 - 2l_{21}l_{22} \cos \theta_1}} \\ \sin \gamma_3 = \frac{d_{33} \cos \frac{\theta_2}{2}}{\sqrt{l_{32}^2 + d_{33}^2 - 2l_{32}d_{33} \sin \frac{\theta_2}{2}}} \\ \sin \gamma_4 = \frac{d_{43} \cos \frac{\theta_2}{2}}{\sqrt{l_{42}^2 + d_{43}^2 + 2l_{42}d_{43} \sin \frac{\theta_2}{2}}} \end{cases} \quad (9)$$

In muscles 3 and 4 (**Figure 2**), we introduce two turning points at the angular bisector of the elbow to design polyline muscles, where d_{33} and d_{43} are the distances from the elbow to the

turning points of muscles 3 and 4, respectively. From Equation (9), it is clear that $\sin \gamma_i$ is a nonlinear function of θ_i . By substituting Equation (9) into Equation (8), we obtain muscle torque functions $\tau'_{n1}(F_1^T, F_2^T, \theta_1)$ and $\tau'_{n2}(F_3^T, F_4^T, \theta_2)$.

For the muscle description in our arm model, it is difficult to determine its inverse function, because F^T and Equation (5) are piecewise functions with complicated expressions. Therefore, we usually cannot calculate u_i by directly using muscle force, but instead we adopt an indirect method.

We assume that expected states $\theta_i, \dot{\theta}_i$ and $\ddot{\theta}_i$ are given. Expected torque τ_n can be calculated by Equation (6) as a learning target. On the other hand, actual tendon force F^T is known when corresponding excitation signals u are generated, and hence actual torque τ'_n is calculable. To obtain actual angular accelerations $\ddot{\theta}_i$, Equation (6) can be computed reversely. In general, $\ddot{\theta}$ can be rewritten as $\ddot{\theta}(\tau_n, \theta, \dot{\theta})$. Considering $\dot{\theta} = \frac{d\theta}{dt}$ and $\ddot{\theta} = \frac{d\dot{\theta}}{dt}$, joint angle θ at time $(t + 1)$ can be obtained as

$$\theta_{t+1}(\tau_n, \theta_t(\ddot{\theta}_{t-1}), \dot{\theta}_t(\ddot{\theta}_{t-1}), \ddot{\theta}_t). \quad (10)$$

If tendon force vector F^T satisfies

$$\tau_n(\theta, \dot{\theta}, \ddot{\theta}) = \tau'_n(F^T, \theta), \quad (11)$$

we can rewrite Equation (11) as

$$\theta_{t+1}(\tau'_n(F_t^T, \theta_t), \theta_t(\ddot{\theta}_{t-1}), \dot{\theta}_t(\ddot{\theta}_{t-1}), \ddot{\theta}_t). \quad (12)$$

The purpose of our framework is to find appropriate excitation signals u to generate tendon forces that satisfy Equation (11). As a result, the expected motions will be generated during exploration. Based on Equation (12), we establish a training framework for the musculoskeletal arm model. When excitation signal u is given, corresponding activation signal a and tendon force F^T can be calculated by muscle dynamics. Then, new motion states can be solved using the arm model. If excitation signal u is unknown, we should explore candidate solutions to generate F^T satisfying (Equation 11).

3. HUMAN-INSPIRED PHASED TARGET LEARNING FRAMEWORK

We design a learning framework to solve signal u_i . Conventional learning frameworks use expected states as the learning target. However, these targets can cause unforeseen problems during the solving process, and solutions can fall into local optima. In contrast, the proposed PTL framework can avoid local optima by guiding the learning process. Specifically, different learning targets are designed according to the learner's level, additionally providing high efficiency during training. We consider the musculoskeletal system, optimization model, and expected target state as the most essential aspects in our framework (Figure 4) and detail the last two parts in the sequel.

3.1. Phased Target Learning

3.1.1. Simplified Target Setup

Consider a beginner who starts to learn dancing or practicing a sport. It is difficult for him to acquire all the professional

postures and skills at once. Instead of trying to enhance memory or learning skills, the simplest solution is reducing the quality requirements and perform intensive practice through gradual improvement. This way, the beginner will easily improve by establishing simple learning targets that are gradually set at different levels as learning proceeds. In this study, we calculated precise motion states to be expected targets. Then, we designed different simplified states as easier targets for learning. Formally, let $s \in \mathcal{S}$ be the expected states of the arm model, and $s_T \in \mathcal{S}_T$ be the simplified states. s_T can be calculated by simplifying s :

$$s_T(t) = s(\text{ceil}(\frac{t}{d}) \cdot d) \cdot \delta(0), t = 1, 2, \dots, T, \quad (13)$$

where $\delta(t)$ is an impulse function, and $d \in \mathbb{N}_+$ satisfying $\frac{d}{T} \in [\frac{1}{T}, 1]$ is a forgetting factor. When $d = T$, s_T only reflects the endpoint state of expected state s , and when $d = 1$, $s_T = s$, indicating that s_T reflects all the states of \mathcal{S} .

Obviously, simplification induces errors with respect to expected states. Suppose that $\theta \in \mathcal{S}$ is the expected joint angle of the arm model, and $\theta_T \in \mathcal{S}_T$ is the simplified joint angle. Then, we define the average allowed error between s and s_T as

$$\bar{e}_T = \frac{1}{T} \sum_{t=1}^T |\theta(t) - \theta_T(t)|. \quad (14)$$

According to Equations (13) and (14), average allowed error \bar{e}_T depends only on the forgetting factor d . Geometrically, \bar{e}_T can be considered as the width of the equivalent error region. Figure 5 shows the width and effect of d on simplified joint angle curve θ_T .

PTL provides different simplified targets for learning at varying training phases. When the motion accuracy achieves the average allowed error range, \bar{e}_T , a new and smaller average allowed error range is given to guide training. Then, we define actual average error \bar{e}_R of motion as

$$\bar{e}_R = \frac{1}{T} \sum_{t=1}^T |\theta_T(t) - \theta_R(t)|. \quad (15)$$

Unlike Equation (14), Equation (15) uses the actual joint angle, θ_R . In addition, \bar{e}_T is updated after each training iteration. A new average allowed error is computed only when

$$\bar{e}_T - \bar{e}_R > 0. \quad (16)$$

$$d = \begin{cases} D(d), & \bar{e}_T - \bar{e}_R > 0, D(d) \geq 1; \\ 1, & \bar{e}_T - \bar{e}_R > 0, D(d) < 1; \\ d, & \bar{e}_T - \bar{e}_R \leq 0; \end{cases} \quad (17)$$

Equation (17) is the update rule of forgetting factor d , where $D(d)$ is a function that satisfies $D(d) < d$. It is convenient to maintain the value of $|d - D(d)|$ small, because a large difference between adjacent simplified states vanishes the gradual learning effect.

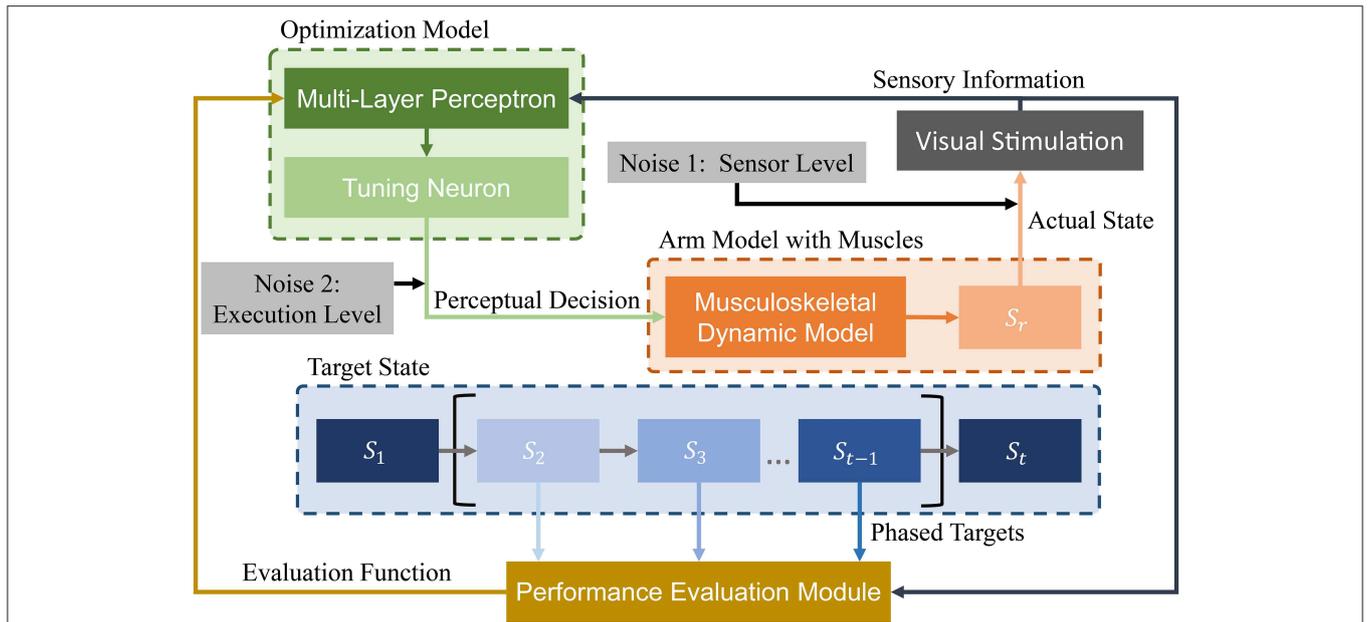


FIGURE 4 | Schematic of proposed PTL framework for motion control of musculoskeletal robots. A vision sensor collects motion information. Then, visual stimuli are transmitted to the optimization model and performance evaluation module. During optimization, state information is processed by a multi-layer perceptron. Then, perceptual decisions (excitation signals) are transmitted to the arm model as optimization results. During performance evaluation, different phased targets are designed to guide arm motion states. Finally, the evaluation results are transmitted to the optimization model for improved decision-making. In addition, two biological noise sources are considered during learning for improved exploration ability.

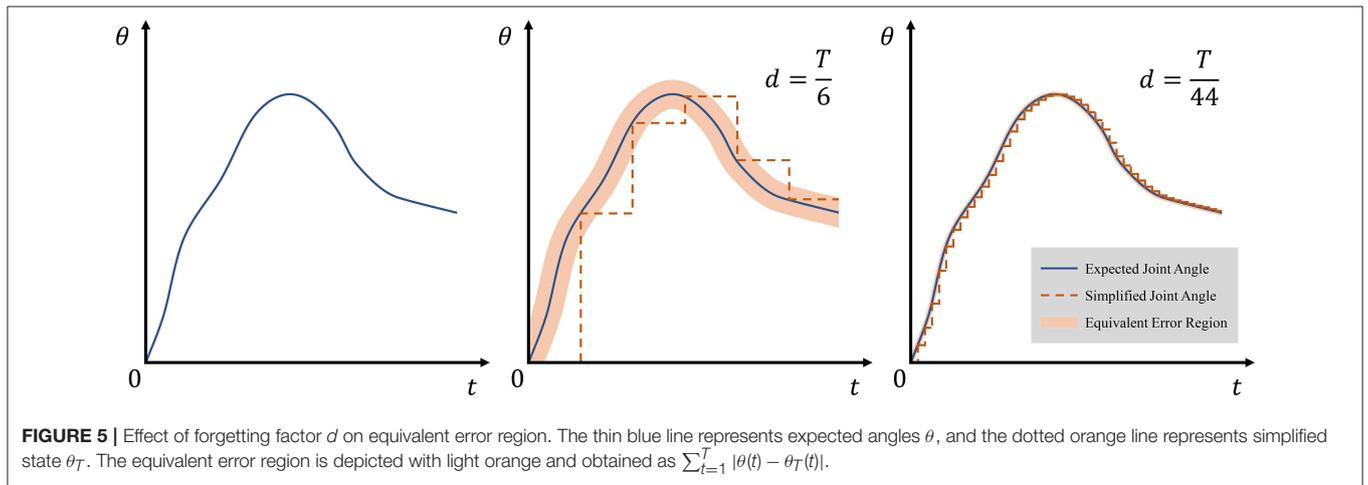


FIGURE 5 | Effect of forgetting factor d on equivalent error region. The thin blue line represents expected angles θ , and the dotted orange line represents simplified state θ_r . The equivalent error region is depicted with light orange and obtained as $\sum_{t=1}^T |\theta(t) - \theta_r(t)|$.

3.1.2. Performance Evaluation Function

Conventional temporal-difference learning methods are highly suitable for model-free learning. Considering Equation (11), the inverse function of τ'_n should be determined and can be set as a model-free problem. In this study, we aimed to improve the Q-network to estimate the continuous excitation signal u for musculoskeletal systems. Then, we combined it with PTL to calculate appropriate control signals.

Let T be the number of finite timesteps and u_i be the excitation signal for muscle i . Each signal $u_i(t)$ at time t has two possible actions; either increase $[a_{i,1}(t)]$ or decrease $[a_{i,2}(t)]$. The

adjustment of u_i affects the muscle and musculoskeletal model at time $(t + 1)$.

However, the two actions only determine the increment sign, and additional parameters are required to calculate the step sizes. Furthermore, the difference between adjacent states can hinder perceptron learning from input states during training. Moreover, incorrect adjustments can lead to signal oscillation in the redundant musculoskeletal model.

In human cortical circuits, sensory information is encoded by neurons via opposite tuning (Romo and de Lafuente, 2013). Based on this mechanism, we redefine action-value function $Q_{u_i,j}$

as a probability of signal u_i executing action $a_{i,j}$. Equation (18) defines u_i as

$$u_i = \frac{1}{\sum_{j=1}^2 Q_{u_i,j}} (Q_{u_i,1} u_{\max} + Q_{u_i,2} u_{\min}), \quad i = 1, 2, \dots, n \quad (18)$$

and action-value function $Q_{u_i,j}$ is redefined as

$$Q_{u_i,j}(s_t, a_{i,j,t}) = \mathbb{E} [E_{u_i}(s_{t+1}, a_{i,j,t+1}) + \gamma Q_{u_i,j}(s_{t+1}, a_{i,j,t+1})], \quad i = 1, 2, \dots, n; \quad j = 1, 2, \quad (19)$$

where E_{u_i} is an evaluation function related to arm motion. According to Equations (18) and (19), a specific action value of a function is not enough to obtain the excitation signal in our method. Instead, relative values of different functions determine an excitation signal, and thus $Q_{u_i,1}$ and $Q_{u_i,2}$ should be maintained balanced. In addition, note that E_{u_i} is used in Equation (19) instead of conventional reward function R_{u_i} . This is because the R_{u_i} is a decreasing function of the action error, and during training, reducing action errors increases R_{u_i} and $Q_{u_i,j}$. In this case, the balance of action-value functions is affected by increasing $Q_{u_i,j}$. Therefore, we employ evaluation function E_{u_i} , which is an increasing function of the action error. Reducing errors therefore imply smaller E_{u_i} and a weaker effect than R_{u_i} on the balance of action-value functions. Furthermore, $(E_{u_i})_{\min} > 0$ promotes stability, as detailed in section 3.1.3.

We obtain the performance evaluation function as follows:

$$E_{u_i}(e_R) = p \cdot \exp [m \cdot g^2(e_R)] + k \quad (20)$$

$$g(e_R) = \min [|e_R|, e_0], \quad (21)$$

where $p, m, k > 0$ are parameters of E_{u_i} and function $g(e_R)$ prevents exploding gradients under large errors.

3.1.3. Learning by Gradient Descent

We define the loss function by summing the squared errors between expected action value $Q'_{u_i,j}$ and actual action value $E_{u_i,j} + \gamma Q'_{u_i,j}$:

$$L(\theta) = \frac{1}{2} \mathbb{E} \left[\sum_{i=1}^n \sum_{j=1}^2 \left(E_{u_i,j} + \gamma Q'_{u_i,j}(s', a'; \theta') - Q_{u_i,j}(s, a; \theta) \right)^2 \right], \quad (22)$$

where γ is a factor to discount the future action value. The gradient of the loss function is given by

$$\nabla L(\theta) = \mathbb{E} \left[\sum_{i=1}^n \sum_{j=1}^2 \gamma \left(E_{u_i,j} + \gamma Q'_{u_i,j}(s', a'; \theta') - Q_{u_i,j}(s, a; \theta) \right) \nabla Q'_{u_i,j}(s', a'; \theta') \right]. \quad (23)$$

During backpropagation, the outputs of multi-layer perceptron in our model can be easily obtained. We suppose that $Q'_{u_i,j}$ represents the result of the output layer and can be expressed as

$$Q'_{u_i,j} = f \left(\sum_{h=1}^{n_h} \omega_{hk} y_h \right), \quad (24)$$

where $f(x)$ is the sigmoid activation function, ω_{hk} is the weight of the edge from the h -th node in the hidden layer to the k -th node in the output layer. Consider ω_{hk} as an example, the weight increment is given by

$$\Delta \omega_{hk} = -\eta \frac{\partial L}{\partial \omega_{hk}} \quad (25)$$

$$= -\eta \sum_{i=1}^n \sum_{j=1}^2 \gamma (E_{u_i,j} + \gamma Q'_{u_i,j} - Q_{u_i,j}) \frac{\partial Q'_{u_i,j}}{\partial \omega_{hk}} \quad (26)$$

$$= -\eta \sum_{i=1}^n \sum_{j=1}^2 \gamma (E_{u_i,j} + \gamma Q'_{u_i,j} - Q_{u_i,j}) f' \left(\sum_{h=1}^{n_h} \omega_{hk} y_h \right) \sum_{h=1}^{n_h} y_h, \quad (27)$$

where y_h is the output of the h -th node in the hidden layer. When the excitations become stable, the expected increment is $\Delta \omega_{hk} \rightarrow 0$ such that $\Delta Q_{u_i,j} \rightarrow 0$, and hence $E_{u_i,j} + \gamma Q'_{u_i,j} = Q_{u_i,j}$ at this time. Factor γ is known as a decimal, and we can infer $\gamma Q'_{u_i,j} < Q_{u_i,j}$, which explains why the performance evaluation function should satisfy $(E_{u_i,j})_{\min} > 0$.

3.2. Noise in Nervous System

Noise is ubiquitous in real-world systems, especially during information transmission. As motion learning consists of information transmission, noise is present. Recent research roughly identified noise sources in the nervous system at the sensor and action levels (A Aldo et al., 2008). We considered these noise sources in the proposed PTL framework.

3.2.1. Noise at Sensor Level

During the collection of visual information, photoreceptors receive photons reflected by objects under the influence of Poisson noise, which reduces the accuracy of optical information (Bialek, 1987). Although sensory noise is inevitable (Bialek and Setayeshgar, 2005), it also mitigates sensitivity of the redundant musculoskeletal system.

When motion tracking is performed on the redundant musculoskeletal arm model, the Q-network method can exhibit unstable training, because joint angles are affected by the action of many muscles, likely falling into local optima. Then, any small fluctuation of excitation signals can be amplified and cause divergent signals. However, when target motion is considered as a region, fluctuations are tolerated. We use Poisson noise to conform tolerance regions and prevent rapid fluctuations:

$$s_{RN} = s_R + N_1, \quad (28)$$

$$N_1 \sim \text{Pois}(\lambda), \quad (29)$$

where s_R is the actual arm state, s_{RN} is the observed arm state observed by the vision sensor, and N_1 is Poisson noise in the visual information. In our algorithm, let $s_R = s_{RN}$ represents the inputs of the improved Q-network.

3.2.2. Noise at Execution Level

Noise at the sensor level is also called planning noise, as it affects decision-making. In addition, execution noise exists and is superimposed on the original decision signals. In fact, execution

noise describes an uncontrollable noise whose standard deviation is linearly related to the mean muscle force (Hamilton et al., 2004; Dhawale et al., 2017) and can be expressed as

$$u_{Ni} = \min[\max[u_i + N_2, 0], 1], \quad (30)$$

where $N_2 \sim N(0, (\nu F^T)^2)$ simulates noise in the motor system periphery, u_i and u_{Ni} are undisturbed and noisy signals from perceptron, respectively, and ν is a scale coefficient of tendon force F^T . Note that the square of νF^T defines the variance of execution noise, and like noise in sensor level, let $u_i = u_{Ni}$ represent the final outputs of the proposed network.

4. SIMULATION EXPERIMENTS

We conducted simulation experiments on the musculoskeletal system model to verify the performance of different algorithms. Moreover, the equilibria of action values are analyzed to explain the learning process of the proposed PTL framework.

4.1. Experimental Setup

As mentioned above, we designed a simplified musculoskeletal arm model to verify and evaluate the proposed learning method. After analyzing its dynamics (Equation 12), a basic control framework is devised. To validate the formulation and analyze performance, optimization should be performed.

In this study, the proposed PTL is applied to a point-to-point motion task with constant angular velocity as temporal-difference learning approach. For a final state of target motion, we calculated midpoints and required constraints using inverse kinematics. Then, we used joint angles as motion states to design the simplified target states. Assuming a constant angular velocity, four types of control strategies were evaluated: (1) Q-network, (2) Q-network with noises, (3) PTL, and (4) PTL with noises. The implemented method including PTL is detailed in Algorithm 1.

We set maximum number of iterations $K = 500$ and number of timesteps $T = 10,000$ to simulate 10 s. All the errors and control signals were recorded at each timestep.

4.2. Results and Analysis

We considered average error $\bar{e} = \frac{1}{T} \sum_{t=1}^T |\theta(t) - \theta_R(t)|$ as a key performance indicator, where $\theta(t)$ is the precise expected joint angle at time t . As \bar{e} reflects the average error, motion performance can be evaluated from this measure.

Figure 6 shows the average error \bar{e} according to iteration k . Clearly, the Q-network method, Q-network with noises, and PTL are trapped at local optima and unstable during training. Still, phased targets improve learning by increasing the randomness of exploration, and noises during training enhance fault tolerance and the exploration ability during control.

Assume that the ratio of action-value functions is convergent to local optimum b_i , which is defined as

$$b_i = \frac{Q_{u_i,1}}{Q_{u_i,2}}. \quad (31)$$

Algorithm 1: PTL with Noises for Motion Learning in Musculoskeletal System.

Require: Given precise motion states $s(t) \in \mathcal{S}$. Initialize parameters: interval d , maximum number of iterations K , excitation signal u_i . Obtain simplified motion state $s_T(t) \in \mathcal{S}_T$ using Equation 13.

- 1: **for** $k=1$ to K **do**
- 2: Compute average allowed error $\bar{e}_T = \frac{1}{T} \sum_{t=1}^T |s(t) - s_T(t)|$
- 3: **if** $\bar{e}_R < \bar{e}_T$ and $k \neq 1$ **then**
- 4: Reduce d gradually ($d \in N_+$, $d_{\max} < T$)
- 5: Set new target states \mathcal{S}_T by simplifying \mathcal{S}
- 6: **end if**
- 7: **for** $t=0$ to T **do**
- 8: Calculate activation signal $a_i(u_i(t))$ and tendon force F_i^T
- 9: Perform motion corresponding to $s_R(t+1)$ caused by F_i^T
- 10: Obtain actual motion error $e_R(t) = |s_T(t) - s_R(t)|$
- 11: Introduce noise at sensor level into motion states via Equation 28. Let $s_R(t) = s_{RN}(t)$ be the inputs of improved Q-network
- 12: Estimate $Q'_{u_i,j}$ by improved Q-network method
- 13: Update weights ω to obtain new action values $Q_{u_i,j}$ via Equation 23
- 14: Obtain signal $u_i(t)$ using Equation 18
- 15: Introduce noise at execution level into excitation signals via Equation 30. Let $u_i(t) = u_{Ni}(t)$ be the outputs of improved Q-network
- 16: **end for**
- 17: **end for**

Then, u_i can be rewritten as

$$u_i = \frac{1}{\sum_{j=1}^2 Q_{u_i,j}} (Q_{u_i,1} u_{\max} + Q_{u_i,2} u_{\min}) \quad (32)$$

$$= \frac{1}{b_i + 1} (b_i u_{\max} + u_{\min}), \quad (33)$$

and hence the equilibrium point b_i is the only parameter that affects excitation signal u_i . We prescribe that the control method adjusts $Q_{u_i,1}$ and $Q_{u_i,2}$ in an opposite way. In addition, increment $\Delta Q_{u_i,j}$ satisfies $\Delta Q_{u_i,j} > 0$ and $\Delta Q_{u_i,j} \ll Q_{u_i,j}$ at simulation onset. The next equilibrium point at time $(t+1)$ is $b'_i = (Q_{u_i,1} \mp \Delta Q_{u_i,1}) / (Q_{u_i,2} \pm \Delta Q_{u_i,2})$, whose increment is given by

$$b_i - b'_i = \frac{Q_{u_i,1}}{Q_{u_i,2}} - \frac{Q_{u_i,1} \mp \Delta Q_{u_i,1}}{Q_{u_i,2} \pm \Delta Q_{u_i,2}}, \quad (34)$$

$$= \frac{\pm(Q_{u_i,1} \Delta Q_{u_i,2} + \Delta Q_{u_i,1} Q_{u_i,2})}{Q_{u_i,2}(Q_{u_i,2} \pm \Delta Q_{u_i,2})}. \quad (35)$$

For $(-\Delta Q_{u_i,1}, +\Delta Q_{u_i,2})$, we obtain $b_i - b'_i > 0$, and excitation signal u_i becomes smaller. For $(+\Delta Q_{u_i,1}, -\Delta Q_{u_i,2})$, as $Q_{u_i,2} - \Delta Q_{u_i,2} > 0$, we obtain $b_i - b'_i < 0$, and excitation signal u_i becomes larger.

However, with reducing motion error, the increment of function $Q_{u_i,j}$ is smaller for $Q_{u_i,j} \approx \Delta Q_{u_i,j}$. From Equation

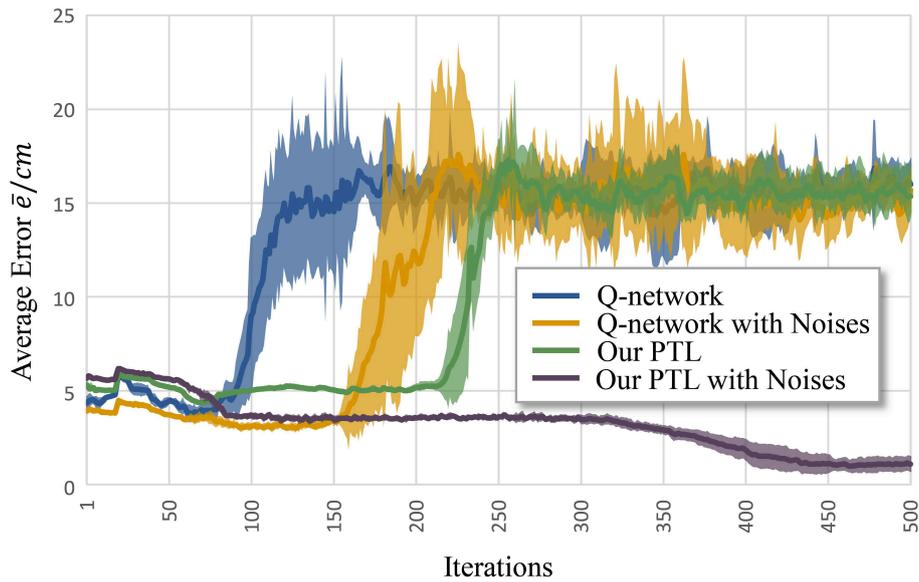


FIGURE 6 | Average error for different methods to control musculoskeletal arm model for motion tracking. Curves correspond to average errors over 10 trials.

(35), when $(+\Delta Q_{u_i,1}, -\Delta Q_{u_i,2})$, the sign of $(b_i - b'_i)$ depends on the sign of $(Q_{u_i,2} - \Delta Q_{u_i,2})$. Nevertheless, it is difficult to guarantee either $(Q_{u_i,2} \leq \Delta Q_{u_i,2})$ or $(Q_{u_i,2} \geq \Delta Q_{u_i,2})$. The uncertain sign causes chattering on the excitation signal (Equation 33), which can cause signal divergence at the final state.

In addition, random factors like ϵ and noise can give rise to fluctuations of $\Delta Q_{u_i,j}$, which may increase the adjustment extent. For example, if $(+\Delta Q_{u_i,1}, +\Delta Q_{u_i,2})$ or $(-\Delta Q_{u_i,1}, -\Delta Q_{u_i,2})$, the increment of b_i is given by

$$b_i - b'_i = \frac{Q_{u_i,1}}{Q_{u_i,2}} - \frac{Q_{u_i,1} \pm \Delta Q_{u_i,1}}{Q_{u_i,2} \pm \Delta Q_{u_i,2}} \quad (36)$$

$$= \frac{\pm(Q_{u_i,1}\Delta Q_{u_i,2} - \Delta Q_{u_i,1}Q_{u_i,2})}{Q_{u_i,2}(Q_{u_i,2} \pm \Delta Q_{u_i,2})}, \quad (37)$$

where $(Q_{u_i,1}\Delta Q_{u_i,2} - \Delta Q_{u_i,1}Q_{u_i,2})$ with an uncertain sign can seriously undermine performance, as it is directly related to the sign of $(b_i - b'_i)$. Furthermore, performance may decay even without condition $Q_{u_i,j} \approx \Delta Q_{u_i,j}$, and the method will be unreliable under its influence. Fortunately, with appropriate training, performance degradation by random effects can almost be eliminated.

Another problem is early convergence during learning. **Figure 7** shows the evolution of the average allowed error. The four evaluated methods terminate searching when reaching different local optima. Generally, premature convergence occurs through the insufficient exploration of solutions. Given its exploration ability, the proposed PTL with noises was guided by simplified targets to avoid premature convergence. This method achieved the lowest error (average $\bar{e} < 0.746\text{cm}$) and the most

advanced learning level throughout repeated experiments.

$$b_i = \frac{Q_{u_i,1}}{Q_{u_i,2}} + \Delta b_i \quad (38)$$

We define Δb_i in Equation (38) as a small increment of the equilibrium point caused by the allowed error e_T . As $\frac{Q_{u_i,1}}{Q_{u_i,2}}$ is not at the expected equilibrium point b_i , $Q_{u_i,j}$ cannot easily generate large fluctuations. According to the analyses above, $\frac{Q_{u_i,1}}{Q_{u_i,2}}$ will converge to the final equilibrium point b_i when $t = T$.

Figure 8 shows signal u_i learned using PTL with noises and the corresponding tendon force, F_i^T . **Figure 9** shows the final position of the arm and joint angles. These results show that the most substantial errors occur at motion onset, and only slight fluctuations remain afterwards. At motion onset, it is reasonable to believe that unexpected muscle forces, especially passive forces of muscles 1 and 3, disturb the force balance. As the simulation proceeds, the arm model returns to a balance state by adjusting u_i . Therefore, PTL extends learning and guides toward the next expected solutions. In addition, the noises foster an extensive exploration of the solution space during training.

To further evaluate PTL framework, we consider point-to-point motion through two scenarios. First, motion begins from a stable position ($\theta_i = 0$) and finishes at another position (**Figure 10**).

When motion starts from a stable position, the next state s_{t+1} does not considerably change if $F^T = 0$. Therefore, the algorithm should not deal with large and rapid fluctuations, and the PTL performance is high. In contrast, in the second scenario, motion starts from an unstable position, and s_{t+1} exhibits a large difference when compared with s_t in the initial period

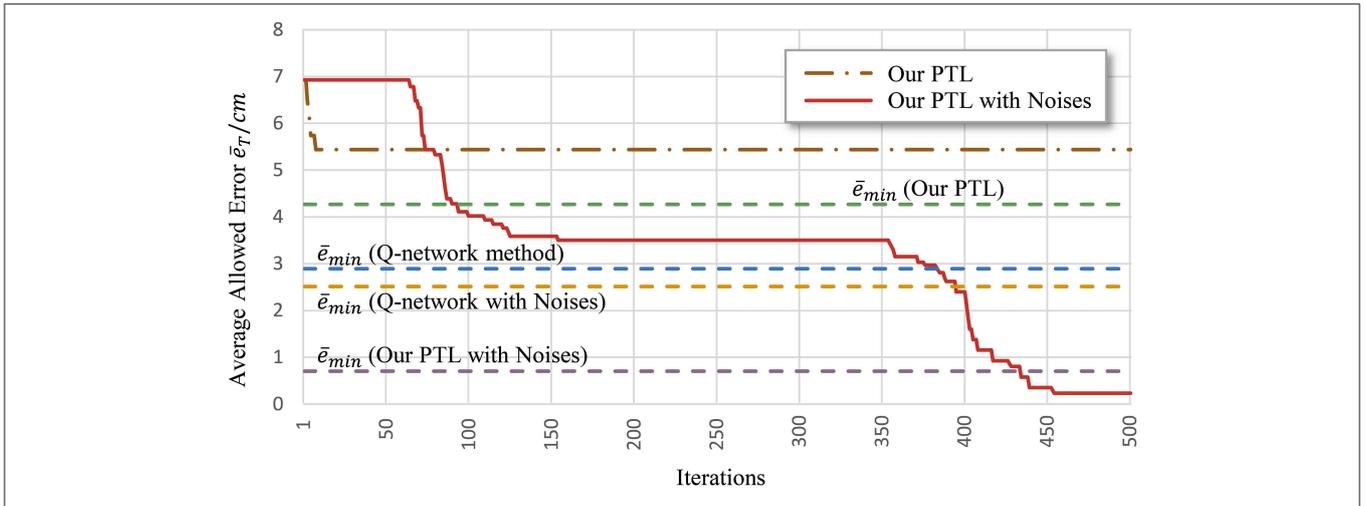


FIGURE 7 | Average allowed error during training. Most algorithms stop learning before processing all the simplified targets.

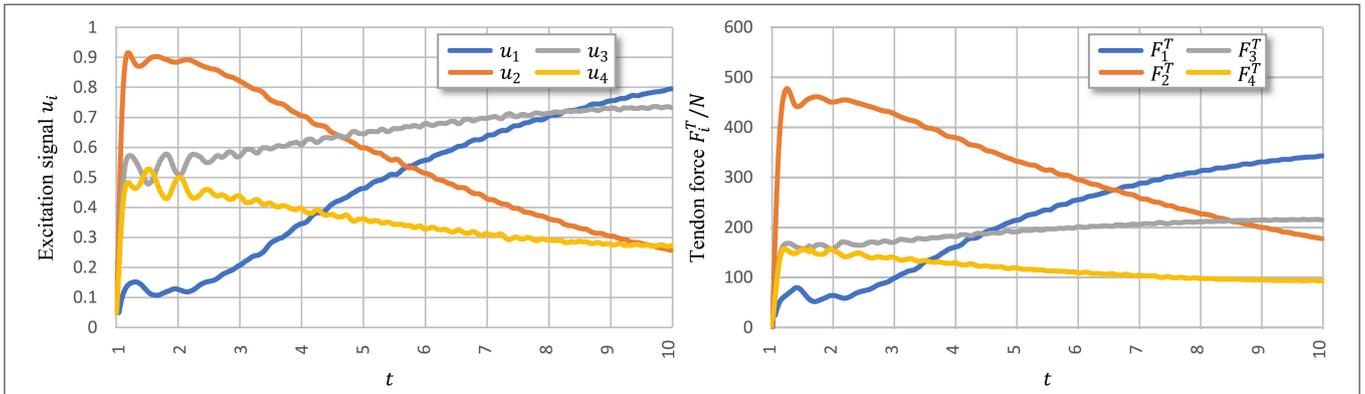


FIGURE 8 | Execution signals trained using PTL with noises after 500 iterations. All excitation signals are filtered with a Butterworth lowpass filter to separate signals from execution noise.

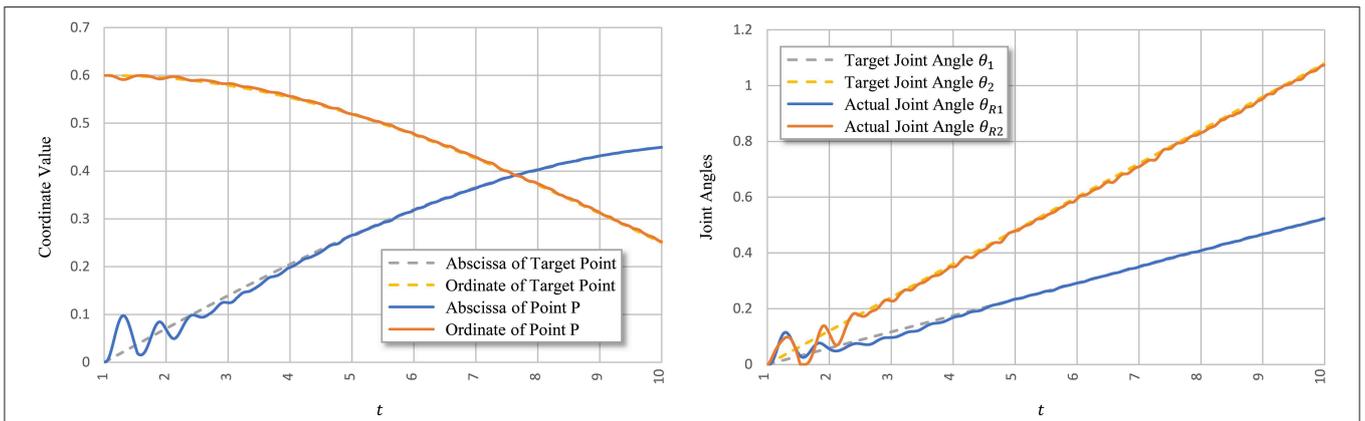


FIGURE 9 | Tracking performance of PTL with noises. Point P is the terminal point for arm motion.

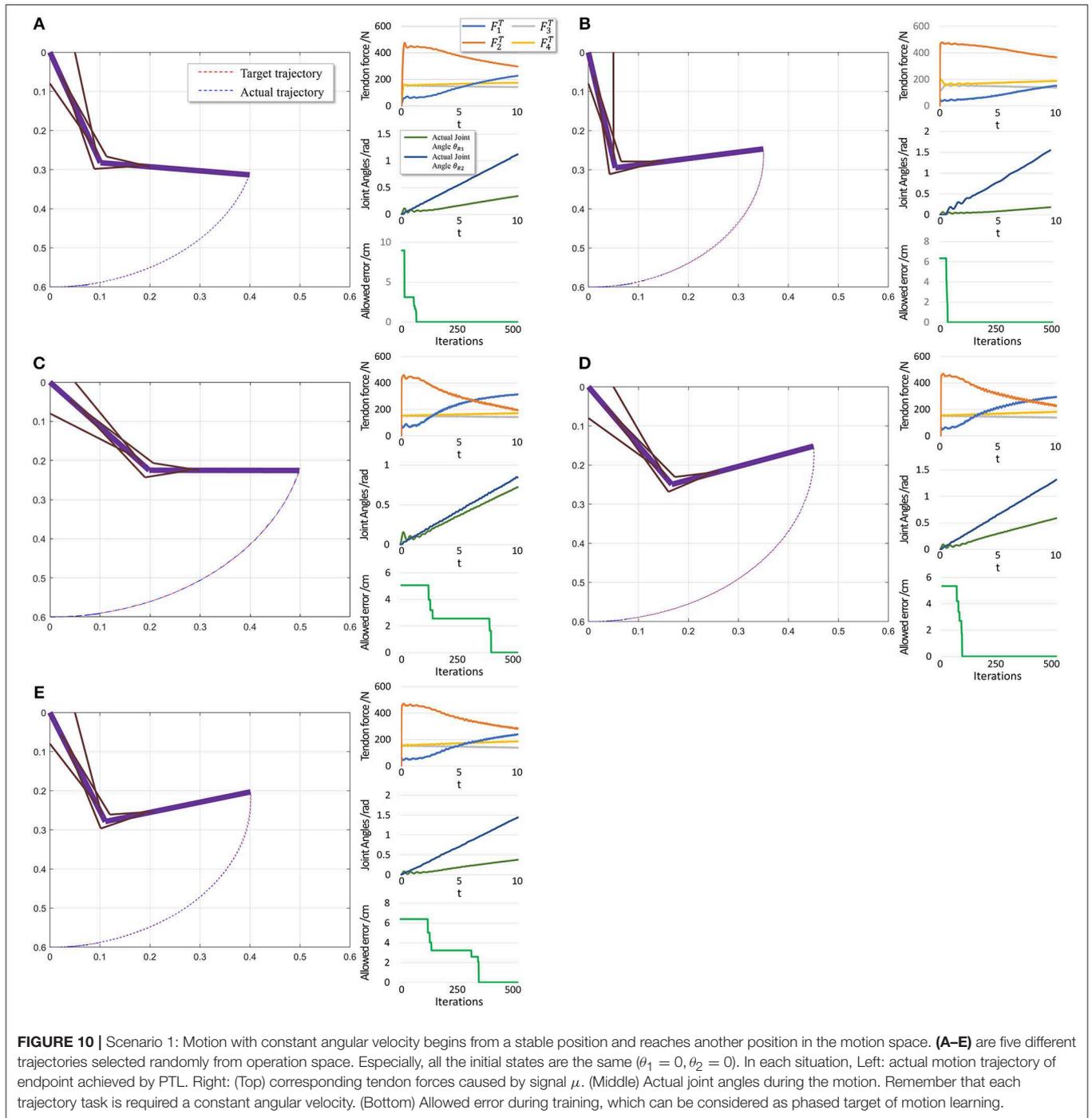


FIGURE 10 | Scenario 1: Motion with constant angular velocity begins from a stable position and reaches another position in the motion space. (A–E) are five different trajectories selected randomly from operation space. Especially, all the initial states are the same ($\theta_1 = 0, \theta_2 = 0$). In each situation, Left: actual motion trajectory of endpoint achieved by PTL. Right: (Top) corresponding tendon forces caused by signal μ . (Middle) Actual joint angles during the motion. Remember that each trajectory task is required a constant angular velocity. (Bottom) Allowed error during training, which can be considered as phased target of motion learning.

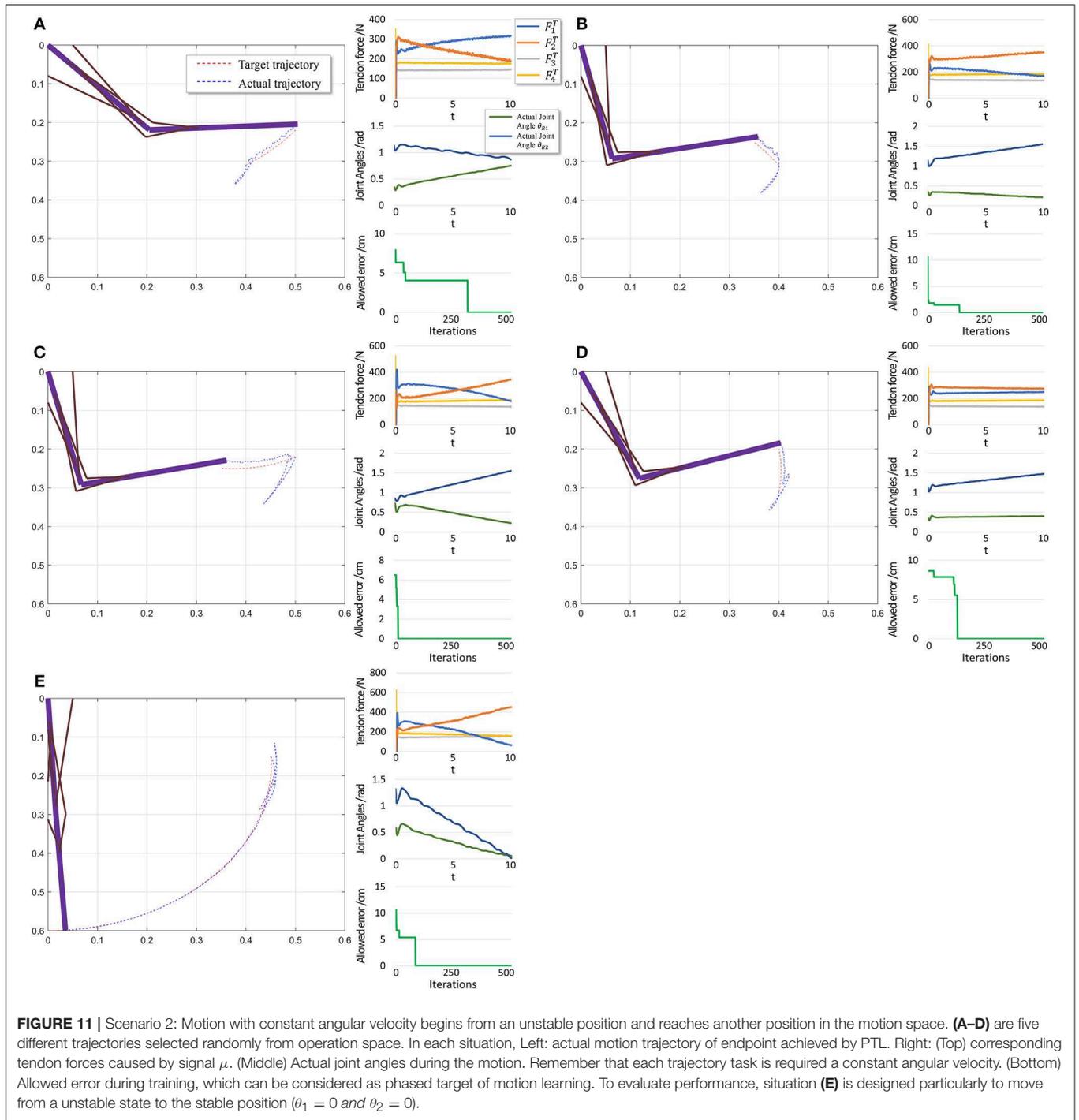
even if $F^T = 0$, as the gravitational torque contributes to a large angular acceleration. Consequently, learning is unstable.

The performance in the second scenario (Figure 11) confirms our prediction of large initial fluctuations. In fact, inappropriate initial parameters in musculoskeletal model will also degrade the performance. As inappropriate parameters lead to inappropriate muscle force, and some timesteps are necessary to adjust those parameters. In addition, the trajectory length is notably shorter

than that in the first scenario, leading to a shorter trajectory for adjustment and learning. Consequently, errors increase in this scenario.

5. CONCLUSIONS

In this paper, we propose a human-inspired motion learning framework for a musculoskeletal system, called PTL. We analyze the learning process and equilibrium point of $Q_{u_{ij}}$,



determining that phased targets guide excitation signals toward expected values during learning. Two types of biological noise sources are considered in the PTL framework to increase the exploration ability in an expanded solution space, making the algorithm suitably follow the guidance of phased targets. Theoretically, as PTL is based on a human learning process, it can be expanded as a general-purpose learning framework if we find appropriate ways to

simplify different kinds of tasks, such as capture and pattern recognition tasks.

In future work, we will apply advanced methods in PTL to improve performance, especially when motion starts from an unstable position. Furthermore, better approaches for simplifying tasks and more biological mechanisms of motion control should be investigated to expand the application scope of the PTL framework.

DATA AVAILABILITY

All datasets generated for this study are included in the manuscript and/or the supplementary files.

AUTHOR CONTRIBUTIONS

JZ provided the main ideas of this research, wrote the manuscript and codes of experiments. JC and HD provided suggestions about PTL framework. HQ and other authors discussed and revised the manuscript.

REFERENCES

- Aldo, F., Selen, L. P. J., and Wolpert, D. M. (2008). Noise in the nervous system. *Nat. Rev. Neurosci.* 9, 292–303. doi: 10.1038/nrn2258
- Agarwal, G. C., Berman, B. M., and Stark, L. (1970). Studies in postural control systems part I: Torque disturbance input. *IEEE Trans. Syst. Sci. Cybern.* 6, 116–121. doi: 10.1109/TSSC.1970.300285
- Arnold, E. M., and Delp, S. L. (2011). Fibre operating lengths of human lower limb muscles during walking. *Philos. Trans. R. Soc. B Biol. Sci.* 366, 1530–1539. doi: 10.1098/rstb.2010.0345
- Asano, Y., Kozuki, T., Ookubo, S., Kawasaki, K., Shirai, T., Kimura, K., et al. (2015). “A sensor-driver integrated muscle module with high-tension measurability and flexibility for tendon-driven robots,” in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Hamburg: IEEE), 5960–5965.
- Asano, Y., Okada, K., and Inaba, M. (2017). Design principles of a human mimetic humanoid: humanoid platform to study human intelligence and internal body system. *Sci. Robot.* 2:eaq0899. doi: 10.1126/scirobotics.aq0899
- Bialek, W. (1987). Physical limits to sensation and perception. *Annu. Rev. Biophys. Biophys. Chem.* 16, 455–478. doi: 10.1146/annurev.bb.16.060187.002323
- Bialek, W., and Setayeshgar, S. (2005). Physical limits to biochemical signaling. *Proc. Natl. Acad. Sci. U.S.A.* 102, 10040–10045. doi: 10.1073/pnas.0504321102
- Chen, J., Zhong, S., Kang, E., and Qiao, H. (2018). Realizing human-like manipulation with musculoskeletal system and biologically inspired control. *Neurocomputing* 339, 116–129. doi: 10.1016/j.neucom.2018.12.069
- Cook, G., and Stark, L. (1968). The human eye-movement mechanism: experiments, modeling, and model testing. *Arch. Ophthalmol.* 79, 428–436. doi: 10.1001/archoph.1968.03850040430012
- Crowninshield, R. D., and Brand, R. A. (1981). A physiologically based criterion of muscle force prediction in locomotion. *J. Biomech.* 14, 793–801. doi: 10.1016/0021-9290(81)90035-X
- Dhawale, A. K., Smith, M. A., and Ölveczky, B. P. (2017). The role of variability in motor learning. *Annu. Rev. Neurosci.* 40, 479–498. doi: 10.1146/annurev-neuro-072116-031548
- Diuk, C., Cohen, A., and Littman, M. L. (2008). “An object-oriented representation for efficient reinforcement learning,” in *Proceedings of the 25th International Conference on Machine Learning (ACM)*, 240–247.
- Eisenberg, E., Hill, T. L., and Chen, Y.-D. (1980). Cross-bridge model of muscle contraction. quantitative analysis. *Biophys. J.* 29, 195–227. doi: 10.1016/S0006-3495(80)85126-5
- Garner, B. A., and Pandy, M. G. (2003). Estimation of musculotendon properties in the human upper limb. *Ann. Biomed. Eng.* 31, 207–220. doi: 10.1114/1.1540105
- Hahn, H. (2013). *Rigid Body Dynamics of Mechanisms: 1 Theoretical Basis*. Springer Science & Business Media.
- Haines, C. S., Lima, M. D., Li, N., Spinks, G. M., Foroughi, J., Madden, J. D., et al. (2014). Artificial muscles from fishing line and sewing thread. *Science* 343, 868–872. doi: 10.1126/science.1246906
- Hamilton, A. F. D. C., Jones, K. E., and Wolpert, D. M. (2004). The scaling of motor noise with muscle strength and motor unit number in humans. *Exp. Brain Res.* 157, 417–430. doi: 10.1007/s00221-004-1856-7
- Hill, A. V. (1938). The heat of shortening and the dynamic constants of muscle. *Proc. R. Soc. Lond. Ser. B Biol. Sci.* 126, 136–195. doi: 10.1098/rspb.1938.0050
- Hou, Y., Liu, L., Wei, Q., Xu, X., and Chen, C. (2017). “A novel ddpg method with prioritized experience replay,” in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (IEEE), 316–321.
- Huxley, A. F. (1957). Muscle structure and theories of contraction. *Prog. Biophys. Biophys. Chem.* 7, 255–318. doi: 10.1016/S0096-4174(18)30128-8
- Huxley, A. F., and Niedergerke, R. (1954). Structural changes in muscle during contraction: interference microscopy of living muscle fibres. *Nature* 173, 971–973. doi: 10.1038/173971a0
- Jagodnik, K. M., and van den Bogert, A. J. (2010). Optimization and evaluation of a proportional derivative controller for planar arm movement. *J. Biomech.* 43, 1086–1091. doi: 10.1016/j.jbiomech.2009.12.017
- Jäntschi, M., Wittmeier, S., Dalamagkidis, K., Panos, A., Volkart, F., and Knoll, A. (2013). “Anthrob-a printed anthropomorphic robot,” in *2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids)* (Atlanta, GA: IEEE), 342–347.
- Kurumaya, S., Suzumori, K., Nabae, H., and Wakimoto, S. (2016). Musculoskeletal lower-limb robot driven by multifilament muscles. *Robomech J.* 3:18. doi: 10.1186/s40648-016-0061-3
- Law, C.-T., and Gold, J. I. (2009). Reinforcement learning can account for associative and perceptual learning on a visual-decision task. *Nat. Neurosci.* 12, 655–663. doi: 10.1038/nn.2304
- Matthew, M., Thomas, U., Ajay, S., and Delp, S. L. (2013). Flexing computational muscle: modeling and simulation of musculotendon dynamics. *J. Biomech. Eng.* 135:021005. doi: 10.1115/1.4023390
- Miriyyev, A., Stack, K., and Lipson, H. (2017). Soft material for soft actuators. *Nat. Commun.* 8:596. doi: 10.1038/s41467-017-00685-3
- Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., et al. (2015). Human-level control through deep reinforcement learning. *Nature* 518, 529–533. doi: 10.1038/nature14236
- Pennestri, E., Stefanelli, R., Valentini, P., and Vita, L. (2007). Virtual musculoskeletal model for the biomechanical analysis of the upper limb. *J. Biomech.* 40, 1350–1361. doi: 10.1016/j.jbiomech.2006.05.013
- Proske, U., and Morgan, D. L. (1987). Tendon stiffness: methods of measurement and significance for the control of movement. A review. *J. Biomech.* 20, 75–82. doi: 10.1016/0021-9290(87)90269-7
- Rasmussen, J., Damsgaard, M., and Voigt, M. (2001). Muscle recruitment by the min/max criterion—a comparative numerical study. *J. Biomech.* 34, 409–415. doi: 10.1016/S0021-9290(00)00191-3
- Riedmiller, M., Gabel, T., Hafner, R., and Lange, S. (2009). Reinforcement learning for robot soccer. *Auton. Robots* 27, 55–73. doi: 10.1007/s10514-009-9120-4
- Romo, R., and de Lafuente, V. (2013). Conversion of sensory signals into perceptual decisions. *Prog. Neurobiol.* 103, 41–75. doi: 10.1016/j.pneurobio.2012.03.007
- Schultz, W., Dayan, P., and Montague, P. R. (1997). A neural substrate of prediction and reward. *Science* 275, 1593–1599. doi: 10.1126/science.275.5306.1593
- Shi, P., and McPhee, J. (2000). Dynamics of flexible multibody systems using virtual work and linear graph theory. *Multibody Syst. Dyn.* 4, 355–381. doi: 10.1023/A:1009841017268

FUNDING

This work was supported in part by the National Key Research and Development Program of China (2017YFB1300200, 2017YFB1300203), the National Natural Science Foundation of China under Grants 91648205 and 61627808, the Strategic Priority Research Program of Chinese Academy of Science under Grant XDB32000000, and the development of science and technology of Guangdong Province special fund project under Grant 2016B090910001.

- Stoianovici, D., and Hurmuzlu, Y. (1996). A critical study of the applicability of rigid-body collision theory. *J. Appl. Mech.* 63, 307–316. doi: 10.1115/1.2788865
- Sutton, R. S., and Barto, A. G. (2018). *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press.
- Tahara, K., and Kino, H. (2010). “Iterative learning scheme for a redundant musculoskeletal arm: Task space learning with joint and muscle redundancies,” in *2010 International Conference on Broadband, Wireless Computing, Communication and Applications* (Fukuoka: IEEE), 760–765.
- Tesauro, G. (1995). Temporal difference learning and td-gammon. *Commun. ACM* 38, 58–69. doi: 10.1145/203330.203343
- Thelen, D. G. (2003). Adjustment of muscle mechanics model parameters to simulate dynamic contractions in older adults. *J. Biomech. Eng.* 125, 70–77. doi: 10.1115/1.1531112
- Thelen, D. G., Anderson, F. C., and Delp, S. L. (2003). Generating dynamic simulations of movement using computed muscle control. *J. Biomech.* 36, 321–328. doi: 10.1016/S0021-9290(02)00432-3
- Van Hasselt, H., Guez, A., and Silver, D. (2016). “Deep reinforcement learning with double Q-learning,” in *Thirtieth AAAI Conference on Artificial Intelligence* (Phoenix, AR).
- Wang, Z., Schaul, T., Hessel, M., Hasselt, H., Lanctot, M., Freitas, N. (2016). “Dueling network architectures for deep reinforcement learning,” in *Proceedings of the 33rd International Conference on Machine Learning* (New York, NY: PMLR), 1995–2003. Available online at: <http://proceedings.mlr.press/v48/wangf16.pdf>
- Winters, J. M. (1990). “Hill-based muscle models: a systems engineering perspective,” in *Multiple Muscle Systems Biomech. & Movem.organiz.* eds J. M. Winters and S. LY. Woo (New York, NY: Springer), 69–93.
- Winters, J. M. (1995). An improved muscle-reflex actuator for use in large-scale neuromusculoskeletal models. *Ann. Biomed. Eng.* 23, 359–374. doi: 10.1007/BF02584437
- Winters, J. M., and Stark, L. (1987). Muscle models: what is gained and what is lost by varying model complexity. *Biol. Cybern.* 55, 403–420. doi: 10.1007/BF00318375
- Zahalak, G. I., and Ma, S.-P. (1990). Muscle activation and contraction: constitutive relations based directly on cross-bridge kinetics. *J. Biomech. Eng.* 112, 52–62. doi: 10.1115/1.2891126
- Zajac, F. E. (1989). Muscle and tendon: properties, models, scaling, and application to biomechanics and motor control. *Crit. Rev. Biomed. Eng.* 17, 359–411.
- Zixing, C. (2000). *Robotics*. Beijing: Tsinghua University Press.
- Conflict of Interest Statement:** The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.
- Copyright © 2019 Zhou, Chen, Deng and Qiao. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.