This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCDS.2020.3045574, IEEE Transactions on Cognitive and Developmental Systems

Motor-Cortex-Like Recurrent Neural Network and Multi-Tasks Learning for the Control of Musculoskeletal Systems

Jiahao Chen, Hong Qiao, *IEEE Fellow*

***Abstract*—The musculoskeletal robot is a promising direction of the next-generation robots. However, current control methods of musculoskeletal robots lack multi-tasks learning ability, great generalization, and biological plausibility. In this article, a motor-cortex-like recurrent neural network (RNN) and a reward modulated multi-tasks learning method are proposed. First, inspired by dynamic system hypothesis of motor cortex, the RNN is introduced to transform movement targets into muscle excitations. The condition that makes a RNN generate motor-cortex-like consistent population response is investigated. Second, a reward modulated multi-tasks learning method of such a RNN is proposed. In the experiments, the control of a musculoskeletal system is realized with multi-tasks learning ability, great generalization, and robustness for noises. Furthermore, the RNN and muscle excitations demonstrate motor-cortex-like consistent population response and human-like muscle synergies respectively. Therefore, the proposed method has better performance and biological plausibility, and verifies the neural mechanisms in the robotic research.**

***Index Terms*—Biologically inspired, Musculoskeletal system, Neuromuscular control, Motor cortex, Muscle synergy, Recurrent neural network**

I. Introduction

Compared with existing joint-link robots, musculoskeletal robots have many superiorities of flexibility, compliance, and robustness. First, redundancy of muscles and joints can realize movements with more flexibility and deal with the failure of actuators. Moreover, the robot can behave compliantly or rigidly with the modulation of muscular co-activation to adapt to different situations. Therefore, many musculoskeletal robots have been established with the imitation of human-like skeleton, bone, joint and muscle [1–21]. For musculoskeletal robots, the muscular modular is the most important component. Most muscular modulars take DC (direct current) motors as power source for their controllability and stability [1–8]. The DC motor drives the lines to contract to generate the movements of skeletons. Furthermore, pneumatic actuators and intelligent materials are also adopted [9–21]. These materials are softer and more similar to the characteristics of biological muscles but have less controllability and stability.

J. Chen, H. Qiao are with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, 100190, China; School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, 100049, China; Beijing Key Laboratory of Research and Application for Robotic Intelligence of Hand-Eye-Brain Interaction, Beijing, 100190, China. *(Corresponding author: Hong Qiao, hong.qiao@ia.ac.cn)*.

H. Qiao is with the CAS Center for Excellence in Brain Science and Intelligence Technology, 200031, Shanghai, China .

Although the musculoskeletal robot has many superiorities, it also brings many challenges of control. The redundancy of robots results in the infinite solutions of muscle excitations. The coupling between muscles and joints makes the individual control of each muscles impossible. In addition, the strong non-linearity of muscular characteristics and sophisticated muscular arrangements make the establishment of explicit mathematic model impossible.

To realize the control of musculoskeletal robots, many methods have been proposed and applied in the hardware platform of musculoskeletal robots [1, 10, 13, 22–27]. With these methods, some simple and imprecise movements can be achieved. Narioka et al. and Ogawa et al. realize the walking task of musculoskeletal robots by the feedforward control with preset activation pattern [10] and feedback intermittent control [13] respectively. Niiyama et al. proposes a motor learning method based on human electromyographic (EMG) to realize running task [22]. Furthermore, some researchers collect robotic data and establish the approximate muscle-joint state mapping with machine learning methods [1, 23–27]. In these methods, the relationship between muscle lengths and joint angles or muscle forces and joint torques is derived and utilized. Then, the robot is controlled by computing the muscle states according to joint states of expected movements. Based on these methods, some human-like free motion, walking, jumping and running tasks with low precision of musculoskeletal robots can be accomplished preliminarily.

To realize more precise movements with the musculoskeletal robot, many novel control methods have been proposed and realized in simulated musculoskeletal systems as proof-of-principle [28–44]. These methods may be further applied to physical musculoskeletal robots in the future. First, some model-based methods have been proposed with establishing explicit mathematical model between the joint space and muscle space of musculoskeletal systems. Based on the model, feedback controller [28], iterative learning controller [29], adaptive controller [30], neuro-fuzzy controller [31], and static optimization [32, 33] are designed to compute muscle excitations or forces. However, the relationship between muscles and joints of sophisticated musculoskeletal robots is complex and difficult to be established explicitly. Therefore, the model-based methods are impossible for the control of sophisticated musculoskeletal robots. Besides, many model-free methods have also been proposed to compute muscle excitations according to movement targets directly. Khan et al. and Nakada et al. train the deep neural network (DNN) through

supervised learning to control the musculoskeletal systems [35, 36]. Furthermore, recurrent neural network (RNN) and DNN are also trained to control the musculoskeletal robots through reinforcement learning methods like reward-based hebbian learning [39], Q learning [40], deep deterministic policy gradient, proximal policy optimization, and trust region policy optimization [37, 38, 41]. In addition, motion primitives are widely used in robotic control [45] and the muscle synergy is a typical primitive in muscle activations. Therefore, some muscle-synergies-based learning have been proposed to compute muscle excitations in reaching task and manipulation task [42–44]. Although these model-free learning methods can be applied to control of sophisticated musculoskeletal robots without establishment of explicit model, the performance of motion generalization is limited and the multi-tasks learning has not been realized.

Human and animals can control musculoskeletal systems to achieve various movement and manipulation tasks with high precision, great generalization and continuous learning ability. Therefore, many scientists focus on understanding how human and animals control the musculoskeletal systems [46–57], which can also stimulate solving the control problem of musculoskeletal robots. Based on the observation of human and animal electromyographic signals, neuroscientists find that muscles are always activated in groups rather than individually. Correspondingly, the hypothesis of muscle synergy is proposed [46–48]. Based on this hypothesis, each group of muscles activated in a time-varying or time-invariant pattern can be regarded as a muscle synergy. Each muscle synergy is modulated with the direction and speed of the movements. Then, muscle excitations are constructed with the combination of multiple modulated muscle synergies. As muscles are stimulated directly by the interneurons and motoneurons in the spinal cord and the neurons in the spinal cord mainly receive the projection from motor cortex, the muscle patterns may be strongly affected by the activities of motor cortex neurons. According to the research of motor cortex, different hypotheses have been proposed to explain the relationship between the neuron activities of motor cortex and movements. Some neuroscientists proposes that the neural states in motor cortex may encode muscle-like commands and control muscles directly [49, 50]. An alternative hypothesis holds that the activities of neurons in motor cortex may encodes kinematic parameters of movements such as direction and speed [51–53]. However, these hypotheses can not explain the neural population dynamics of motor cortex observed in recent physiological investigation. Therefore, Churchland et al. propose a novel assumption that the motor cortex is a dynamic system and the neural states obey smooth dynamics [54–56]. Based on this assumption, the neural states of motor cortex encode both non-muscle-like and muscle-like patterns. Non-muscle-like signals are dominant and muscle-like commands are embedded in an untangled population response.

Inspired by muscle synergy hypothesis and dynamic encoding hypothesis, a novel neuromuscular control method is proposed. The contributions are listed as follows:

1) A pattern of consistent population response of recurrent neural network (RNN) is proposed and the condition that makes RNN generate such consistent population response is investigated with Lyapunov analysis.
2) Furthermore, a reward modulated multi-tasks learning method of such a RNN is proposed with the combination and improvement of orthogonal weight modification and node-perturbation learning.
3) This article also promotes the integration of neuroscience and robotics through validating the dynamic system hypothesis of motor cortex and muscle synergy hypothesis in the control of a musculoskeletal system.

In the experiments, the neuromuscular control of a sophisticated musculoskeletal system is realized through continuous reward modulated learning. The motion learning demonstrates great generalization and robustness for noises. The learned RNN demonstrates motor-cortex-like consistent population response and the muscle excitations generated by RNN shows human-like muscle synergies, which demonstrates the biological plausibility.

The rest of this paper is organized as follows: Section III introduces the proposed method in details. Section IV introduces the experiments and validates the effectiveness of the proposed method. Section V discusses the comparison with relevant work and the improvement of this paper. The conclusion is given in Section VI.

II. Musculoskeletal System

In this section, the dynamics of a musculoskeletal system and the generation of muscle forces are introduced. First, the movement of a musculoskeletal system is driven by muscles as follows:

$$\ddot{\boldsymbol{q}} = \boldsymbol{M}^{-1}(\boldsymbol{q})[\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q}) + \boldsymbol{\tau}] \tag{1}$$

$$\boldsymbol{\tau} = \boldsymbol{W}(\boldsymbol{q})\boldsymbol{F} \tag{2}$$

where $\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}$ are joint angles, velocities, and accelerations respectively. $\boldsymbol{M}$ is the mass matrix, $\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}}$ denote the centripetal and coriolis forces, and $\boldsymbol{G}(\boldsymbol{q})$ is the gravitational force. $\boldsymbol{\tau}$ and $\boldsymbol{F}$ denote joint torques and muscle forces. $\boldsymbol{W}(\boldsymbol{q})$ records the redundant and coupling relationship between muscles and joints. Vectors and matrices are denoted by bold symbols in this paper.

Each muscle consists of muscle fibers and tendons. The muscle fibers can contract to generate active forces under neural excitations and also be stretched to generate passive forces. The forces of muscle fibers are exerted to skeletons through muscle tendons, which connect muscle fibers and skeletons. According to the Hill-type equilibrium muscle model [58, 59], the generation of muscle forces can be represented as follows:

$$\begin{aligned}\boldsymbol{F} &= \boldsymbol{F}_t(\boldsymbol{l}_t) \\ &= \boldsymbol{F}_f(\boldsymbol{a}, \boldsymbol{l}_f, \dot{\boldsymbol{l}}_f,) \otimes cos\boldsymbol{\alpha} \\ &= [\boldsymbol{a} \otimes \boldsymbol{F}_{fl}(\boldsymbol{l}_f) \otimes \boldsymbol{F}_{fv}(\dot{\boldsymbol{l}}_f) + \boldsymbol{F}_{fp}(\boldsymbol{l}_f)] \otimes cos\boldsymbol{\alpha}\end{aligned} \tag{3}$$

where $\boldsymbol{F}_t$ and $\boldsymbol{F}_f$ are forces of muscle tendons and fibers. $\boldsymbol{\alpha}$ denotes the pennation angles between tendons and fibers. $\boldsymbol{a} \otimes \boldsymbol{F}_l(\boldsymbol{l}_f) \otimes \boldsymbol{F}_v(\dot{\boldsymbol{l}}_f)$ and $\boldsymbol{F}_p(\boldsymbol{l}_f)$ denote the active forces and

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCDS.2020.3045574, IEEE Transactions on Cognitive and Developmental Systems

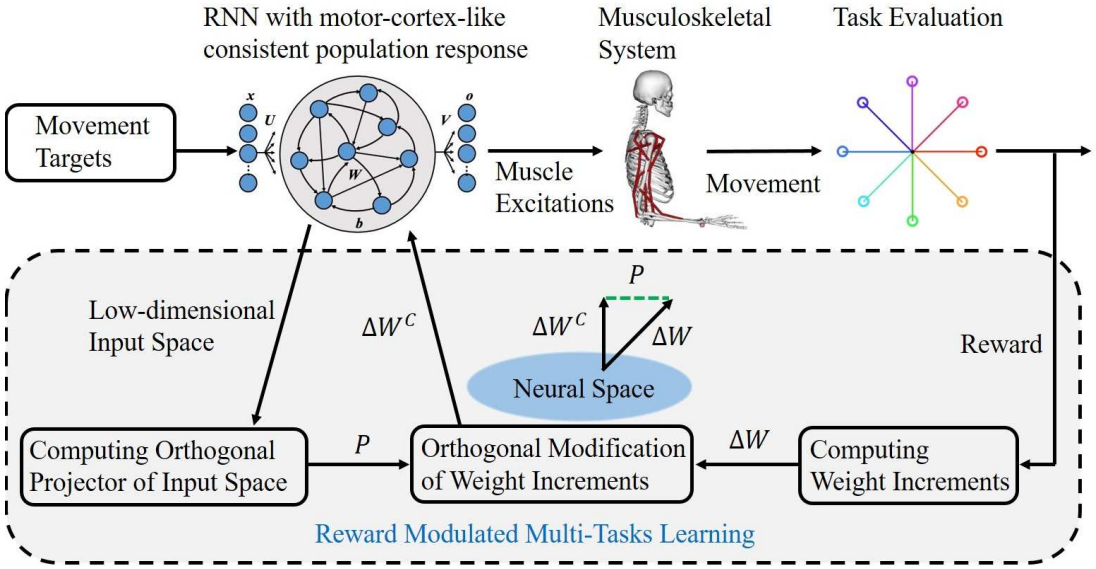

Fig. 1: Framework of multi-tasks learning for the control of musculoskeletal systems

passive forces of muscle fibers. $\boldsymbol{l}_t, \boldsymbol{l}_f, \dot{\boldsymbol{l}}_f$ are tendon lengths, fiber lengths, and fiber velocities. $\boldsymbol{a}$ is the activations of muscle fibers. $\otimes$ denotes elementwise multiplication. The concrete representations of $\boldsymbol{F}_l(\boldsymbol{l}_f)$, $\boldsymbol{F}_v(\dot{\boldsymbol{l}}_f)$, $\boldsymbol{F}_p(\boldsymbol{l}_f)$, and $\boldsymbol{F}_t(\boldsymbol{l}_t)$ are strongly non-linear and can refer to the work in [58].

Each muscle is activated under neural excitations and can be described as follows:

$$\dot{a} = \frac{u - a}{\tau_a(a, u)} \tag{4}$$

$$\tau_a(a, u) = \begin{cases} \tau_{act}(0.5 + 1.5a) & u > a \\ \frac{\tau_{deact}}{0.5 + 1.5a} & u \leq a \end{cases} \tag{5}$$

where a denotes the activation level of all fibers in a muscle and $\dot{a}$ is the derivate of a. u is the neural excitation of a muscle. τ_{act} and τ_{deact} are constants of activation and deactivation respectively.

III. Method

In this section, a motor-cortex-like recurrent neural network (RNN) and a reward modulated multi-tasks learning method are proposed, whose framework is shown in the Fig. 1. First, a pattern of consistent population response of RNN is proposed and the condition is investigated with the Lyapunov analysis. Second, the reward-modulated learning is applied to train the RNN in each task. Furthermore, a multi-tasks learning method of such RNN is proposed with the improvement of the orthogonal weight modification.

A. RNN with Consistent Population Response

1) Consistent Population Response of Motor Cortex: Neurons in motor cortex have strong synaptic connections with the motoneurons in spinal cord and the firing rate of neurons in motor cortex will affect the the contraction of muscles. Based on many investigations [54–56], the motor cortex constitutes a dynamic system and generates motor commands, which can be expressed with a simple deterministic form roughly as follows:

$$\dot{\boldsymbol{r}} = f(\boldsymbol{r}) + \boldsymbol{x} \tag{6}$$

where $\boldsymbol{r}$ is a vector recording firing rates of all neurons in motor cortex, $\dot{\boldsymbol{r}}$ is the derivate, $\boldsymbol{x}$ is the external input, and f is a function describing the dynamic characteristics. In this conception, the neural responses are modulated by the external input but should reflect underlying dynamic characteristics in population level. Through the analysis of neural population during reaching task, the single-neuron responses are disordered but the population response indeed demonstrate consistent population response under different movements. Specifically, with the principal component analysis (PCA) or jPCA method, neural firing rates can be projected into two-dimensional space. The reduced neural states under different movements demonstrate rotational tendency in the same direction.

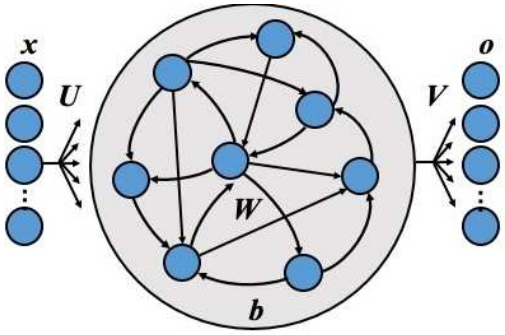

Fig. 2: Architecture of the recurrent neural network

2) Design of RNN: To imitate the neural encoding of motor cortex, recurrent neural network (RNN) is implemented as a classical dynamic system in this paper. The architecture of the RNN is shown in the Fig. 2. The hidden neurons of the RNN adopt leaky neurons and they are fully connected to each other.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCDS.2020.3045574, IEEE Transactions on Cognitive and Developmental Systems

The states of leaky neurons follow the dynamic equations as follows:

$$\tau\dot{\boldsymbol{r}} = -\boldsymbol{r} + \boldsymbol{U}\boldsymbol{x} + \boldsymbol{W}\boldsymbol{h} + \boldsymbol{b} \tag{7}$$

where $\boldsymbol{h} = tanh(\boldsymbol{r})$, $\boldsymbol{r} \in \mathbb{R}^N$ is the vector of membrane potentials of hidden neurons and $\dot{\boldsymbol{r}}$ is the derivate of $\boldsymbol{r}$. $\boldsymbol{h} \in \mathbb{R}^N$ is the vector of firing rates of hidden neurons, $\boldsymbol{x} \in \mathbb{R}^d$ is the vector of external inputs, $\boldsymbol{b} \in \mathbb{R}^N$ is the vector of bias values, $\boldsymbol{U} \in \mathbb{R}^{N\times d}$ is the matrix of input weights from input neurons to hidden neurons, $\boldsymbol{W} \in \mathbb{R}^{N\times N}$ is the matrix of recurrent weights among hidden neurons, and $\tanh(a) = \frac{e^a - e^{-a}}{e^a + e^{-a}}$ is the activation function of hidden neurons.

Then, the output of the RNN is computed as follows:

$$\boldsymbol{o} = ReLU(\boldsymbol{V}\boldsymbol{h}) \tag{8}$$

where $\boldsymbol{o} \in \mathbb{R}^M$ is the vector of outputs, $\boldsymbol{V} \in \mathbb{R}^{M\times N}$ is the matrix of output weights from hidden neurons to output neurons, and $ReLU(a) = max(0, a)$ is the activation function of output neurons.

In order to realize motor-cortex-like consistent population response in the RNN, a pattern of consistent population response, the change rates of all neural states converging to zeros uniformly, is proposed. Then, a Lyapunov function $V(\dot{\boldsymbol{r}})$ is designed to analyze how to realize this pattern as follows:

$$V(\dot{\boldsymbol{r}}) = \dot{\boldsymbol{r}}^T\dot{\boldsymbol{r}} \tag{9}$$

where $V(\dot{\boldsymbol{r}}) > 0$ for $\forall\dot{\boldsymbol{r}} \neq 0$, and $\dot{\boldsymbol{r}} = \frac{1}{\tau}[-\boldsymbol{r} + \boldsymbol{U}\boldsymbol{x} + \boldsymbol{W}\tanh(\boldsymbol{r}) + \boldsymbol{b}]$.

Then, the derivate of the Lyapunov function $V(\dot{\boldsymbol{r}})$ with regard to time is derived as follows:

$$\begin{aligned}\dot{V}(\dot{\boldsymbol{r}}) &= (\tfrac{\partial\dot{\boldsymbol{r}}}{\partial t})^T\dot{\boldsymbol{r}} + \dot{\boldsymbol{r}}^T(\tfrac{\partial\dot{\boldsymbol{r}}}{\partial t}) \\ &= (\tfrac{\partial\dot{\boldsymbol{r}}}{\partial\boldsymbol{r}}\dot{\boldsymbol{r}})^T\dot{\boldsymbol{r}} + \dot{\boldsymbol{r}}^T(\tfrac{\partial\dot{\boldsymbol{r}}}{\partial\boldsymbol{r}}\dot{\boldsymbol{r}}) \\ &= \dot{\boldsymbol{r}}^T[(\tfrac{\partial\dot{\boldsymbol{r}}}{\partial\boldsymbol{r}})^T + (\tfrac{\partial\dot{\boldsymbol{r}}}{\partial\boldsymbol{r}})]\dot{\boldsymbol{r}} \\ &= \tfrac{1}{\tau}\dot{\boldsymbol{r}}^T[(\boldsymbol{W}\tfrac{\partial\tanh(\boldsymbol{r})}{\partial\boldsymbol{r}} - \boldsymbol{I})^T + (\boldsymbol{W}\tfrac{\partial\tanh(\boldsymbol{r})}{\partial\boldsymbol{r}} - \boldsymbol{I})]\dot{\boldsymbol{r}}\end{aligned} \tag{10}$$

As $0 \leq \frac{\partial\tanh(r_i)}{\partial r_i} \leq 1$ for $\forall r_i$ in $\boldsymbol{r}$, it can be obtained that $w\frac{\partial\tanh(r_i)}{\partial r_i} \leq |w|$. Therefore, it can be derived as follows:

$$\boldsymbol{W}\frac{\partial\tanh(\boldsymbol{r})}{\partial\boldsymbol{r}} \leq \boldsymbol{W}_+ \tag{11}$$

where $\frac{\partial\tanh(\boldsymbol{r})}{\partial\boldsymbol{r}} = diag(\frac{\partial\tanh(r_1)}{\partial r_1}, ..., \frac{\partial\tanh(r_N)}{\partial r_N}) \in \mathbb{R}^{N\times N}$ is a diagonal matrix, each element w_{ij}^+ in $\boldsymbol{W}_+$ is the absolute value of the corresponding w_{ij} in $\boldsymbol{W}$.

Then, substituting Eq.11 into Eq.10, it can be obtained that,

$$\begin{aligned}\dot{V}(\dot{\boldsymbol{r}}) &\leq \tfrac{1}{\tau}\dot{\boldsymbol{r}}^T(\boldsymbol{W}_+^T + \boldsymbol{W}_+ - 2\boldsymbol{I})\dot{\boldsymbol{r}} \\ &= \tfrac{1}{\tau}\dot{\boldsymbol{r}}^T(\boldsymbol{P}\boldsymbol{D}\boldsymbol{P}^T - 2\boldsymbol{P}\boldsymbol{P}^T)\dot{\boldsymbol{r}} \\ &= \tfrac{1}{\tau}\dot{\boldsymbol{r}}^T[\boldsymbol{P}(\boldsymbol{D} - 2\boldsymbol{I})\boldsymbol{P}^T]\dot{\boldsymbol{r}} \\ &= \tfrac{1}{\tau}\textstyle\sum_i(\lambda_i - 2)(\dot{\boldsymbol{r}}^T\boldsymbol{P}_{:,i})^2\end{aligned} \tag{12}$$

where $\boldsymbol{W}_+^T + \boldsymbol{W}_+$ is a real and symmetrical matrix and it can be decomposed into $\boldsymbol{P}\boldsymbol{D}\boldsymbol{P}^T$. $\boldsymbol{P} \in \mathbb{R}^{N\times N}$ is an orthogonal matrix and $\boldsymbol{P}\boldsymbol{P}^T = \boldsymbol{I}$. $\boldsymbol{D} = diag(\lambda_1, ..., \lambda_N) \in \mathbb{R}^{N\times N}$ is a diagonal matrix and $\lambda_1, ..., \lambda_N$ are eigenvalues of $\boldsymbol{W}_+^T + \boldsymbol{W}_+$.

Based on the Eq. 12, $\dot{V}(\dot{\boldsymbol{r}}) < 0$ holds for $\forall\dot{\boldsymbol{r}}$ if $\lambda_i < 2$ for $\forall i$. As $\boldsymbol{W}_+^T + \boldsymbol{W}_+$ is an non-negative matrix, the constraint of $\lambda_i < 2$ for $\forall i$ equals to $\rho(\boldsymbol{W}_+^T + \boldsymbol{W}_+) < 2$. When $\rho(\boldsymbol{W}_+^T + \boldsymbol{W}_+) < 2$ is satisfied strictly, the change rates $\dot{\boldsymbol{r}}$ of neural states converge to zero uniformly under any time-invariant external inputs and the RNN has the consistent population response. However, the $||\boldsymbol{W}||_F$ is very small under this condition. The RNN with very small $||\boldsymbol{W}||_F$ has poor representation ability and cannot characterize the complex relationship between movement targets and muscle commands. Therefore, a contradiction between the consistent population response and representation ability of RNN exists. In order to guarantee both of the consistent population response and enough representation ability, the parameter $\boldsymbol{b}$ should also be designed appropriately. Taking the one-dimensional RNN ($\tau\dot{r} = -r + ux + wh + b$) with time-invariant external input as an example, the design of parameter b is analyzed. The corresponding Lyapunov function and its derivative are $V(\dot{r}) = \dot{r}^2$ and $\dot{V}(\dot{r}) = [2w\frac{d\tanh(r)}{dr} - 2]\dot{r}^2$ respectively. When $w < 1$, the above constraint is satisfied and $\dot{V}(\dot{r}) < 0$ always holds. Under this condition, the RNN has global consistent population response. When $w > 1$, the above constraint is violated and $\dot{V}(\dot{r}) < 0$ does not hold at the neighborhood of the origin ($r = 0$) but still holds when r is far away from the origin for $0 \leq \frac{d\tanh(r)}{dr} \leq 1$ and $\frac{d\tanh(r)}{dr}|_{r=0} = 1$. Under this condition, the RNN only has local consistent population response. As enough big b or $ux + b$ can confine the r to the local region far away from the origin, RNN can always operates in this local area and has consistent population response. Generalizing the analysis to the high dimensional RNN, improving $||\boldsymbol{b}||_F$ can alleviate the instability caused by the improvement of $||\boldsymbol{W}||_F$. Based on above proof and analysis, $\boldsymbol{W}$ is designed based on the constraint of $\rho(\boldsymbol{W}_+^T + \boldsymbol{W}_+) < 2$ and can violate this strict constraint to obtain enough representation ability. Correspondingly, the $\boldsymbol{b}$ should also be designed with enough big $||\boldsymbol{b}||_F$ to guarantee the consistent population response.

B. Reward Modulated Learning

In order to realize neuromuscular control, the RNN should transform movement targets into muscle excitations. In this section, the RNN is trained with the reward modulated learning.

Although the RNN is introduced as continuous form in the Section III-A, the RNN is performed with the discrete form in practice as follows:

$$\boldsymbol{r}_t = (1-\alpha)\boldsymbol{r}_{t-1} + \alpha(\boldsymbol{U}\boldsymbol{x} + \boldsymbol{W}\boldsymbol{h}_{t-1} + \boldsymbol{b}) \tag{13}$$

where $\boldsymbol{r}_t \in \mathbb{R}^N$, $\boldsymbol{h}_t \in \mathbb{R}^N$ are the vectors of membrane potentials and firing rates of hidden neurons respectively, α is a decay factor of membrane potentials.

During the training phase, some perturbations are applied to improve the randomness of outputs and the exploration of learning. For example, the membrane potentials are perturbed with gaussian noises $\boldsymbol{\epsilon}_t$ as follows:

$$\begin{aligned}\boldsymbol{r}_t^\epsilon &= \boldsymbol{r}_t + \boldsymbol{\epsilon}_t \\ &= (1-\alpha)\boldsymbol{r}_{t-1} + \alpha(\boldsymbol{U}\boldsymbol{x} + \boldsymbol{W}\boldsymbol{h}_{t-1} + \boldsymbol{b}) + \boldsymbol{\epsilon}_t\end{aligned} \tag{14}$$

where $\boldsymbol{r}_t^\epsilon$ is the vector of perturbed membrane potentials and $\boldsymbol{\epsilon}_t \sim N(\boldsymbol{0}, \Sigma) \in \mathbb{R}^N$ is the vector of noises, $\Sigma =$

$diag(\sigma^2, ..., \sigma^2) \in \mathbb{R}^{N\times N}$ is the diagonal covariance matrix of the normal distribution, and σ^2 is the variance of noise. Then, the membrane potential of a hidden neuron at each time step can be regarded as a random variable drawn from a Gaussian distribution as follows:

$$r_i^\epsilon(t) \sim N(r_i(t), \sigma^2) \tag{15}$$

Therefore, the probability density function $g_i(t)$ of each hidden neuron at each time step is represented as follows:

$$\begin{aligned} g_i(t) &= P(r_i^\epsilon(t)|r_i(t), \sigma) \\ &= \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{[r_i^\epsilon(t)-r_i(t)]^2}{2\sigma^2}} \end{aligned} \tag{16}$$

Based on REINFORCE algorithm, each weight can be updated after an episode, namely a movement, as follows:

$$\begin{aligned} \Delta w_{ij} &= \beta(R-\bar{R}) \textstyle\sum_{t=1}^{T_t} e_{ij}(t) \\ &= \beta(R-\bar{R}) \textstyle\sum_{t=1}^{T_t} \frac{\partial ln\, g_i(t)}{\partial w_{ij}} \\ &= \beta(R-\bar{R}) \textstyle\sum_{t=1}^{T_t} \frac{\partial ln\, g_i(t)}{\partial r_i(t)} \frac{\partial r_i(t)}{\partial w_{ij}} \\ &= \beta(R-\bar{R}) \textstyle\sum_{t=1}^{T_t} \frac{(r_i^\epsilon(t)-r_i(t))}{\sigma^2} \alpha h_j(t-1) \\ &= \frac{\beta\alpha}{\sigma^2}(R-\bar{R}) \textstyle\sum_{t=1}^{T_t} \epsilon_i(t) h_j(t-1) \\ &= \eta(R-\bar{R}) \textstyle\sum_{t=1}^{T_t} \epsilon_i(t) h_j(t-1) \end{aligned} \tag{17}$$

where Δw_{ij} is the increment of w_{ij}, $e_{ij} = \frac{\partial ln\, g_i(t)}{\partial w_{ij}}$ is the characteristic eligibility of w_{ij}, $0 < \eta < 1$ is the learning rate, R is the reward at each episode, and each episode has T_t time steps. $\bar{R}$ is the baseline of rewards and can be regarded as the expectation of future reward, which is computed as follows:

$$\bar{R}_{n+1} = \alpha^R \bar{R}_n + (1-\alpha^R) R_n \tag{18}$$

where n represents the n^{th} episode and $0 < \alpha^R < 1$ is the filter factor.

Therefore, all recurrent weights can be updated after each episode as follows:

$$\Delta \boldsymbol{W} = \eta(R-\bar{R}) \textstyle\sum_{t=1}^{T_t} \boldsymbol{\epsilon}_t \boldsymbol{h}_{t-1}^T \tag{19}$$

Similarly, the input weights and bias can be updated after each episode as follows:

$$\Delta \boldsymbol{U} = \eta(R-\bar{R}) \sum_{t=1}^{T_t} \boldsymbol{\epsilon}_t \boldsymbol{x}_t^T \tag{20}$$

$$\Delta \boldsymbol{b} = \eta(R-\bar{R}) \sum_{t=1}^{T_t} \boldsymbol{\epsilon}_t \tag{21}$$

During the training, the RNN should also maintain the aforementioned consistent population response. Therefore, the RNN is initialized to have consistent population response first. Then, each update of recurrent weights is constrained in a small neighborhood to maintain the expected dynamics of as follows:

$$\Delta \boldsymbol{W} = \begin{cases} \frac{g}{||\Delta \boldsymbol{W}||_F} \Delta \boldsymbol{W} & ||\Delta \boldsymbol{W}||_F > g \\ \Delta \boldsymbol{W} & ||\Delta \boldsymbol{W}||_F \leq g \end{cases} \tag{22}$$

where $g > 0$ is a constant value to constrain the norm of weight update, $||\cdot||_F$ is the Frobenius norm of matrix.

In order to improve the efficiency of learning process, some hyper-parameters like η, α_R, and σ^2 are modulated during learning phase adaptively as follows:

$$\begin{aligned} \eta_n &= \gamma_\eta \exp(-\tfrac{\phi_n}{\tau_p}) \\ \alpha_n^R &= \gamma_{\alpha^R} \exp(-\tfrac{\phi_n}{\tau_p}) \\ \sigma_n^2 &= \gamma_{\sigma^2} \exp(-\tfrac{\phi_n}{\tau_p}) \end{aligned} \tag{23}$$

where n represents the n^{th} episode during training, τ_p is a decay coefficient. γ_η, γ_{α^R}, and γ_{σ^2} are the initial values of η, α^R, and σ^2 respectively. ϕ_n is a factor designed to measure the performance of learning in the n^{th} episode. ϕ_n is computed based on [39] and increases with the improvement of performance gradually.

C. Reward Modulated Multi-Tasks Learning

Human can learn multiple tasks continuously without forgetting previously learned knowledge. To realize continuous multi-tasks learning with RNN, Orthogonal Weight Modification (OWM) method [60] is introduced and improved in this section.

In order to protect the previously learned knowledge, the OWM method updates weights only in the direction orthogonal to the space of previously trained inputs as follows:

$$\Delta \boldsymbol{W}_l' = \kappa \Delta \boldsymbol{W}_l^{BP} \boldsymbol{P}_l \tag{24}$$

where $\Delta \boldsymbol{W}_l'$ is the increment of weights in the l^{th} layer based on the OWM method, $\Delta \boldsymbol{W}_l$ is the increment of weights computed by backpropagation method in supervised learning, $\boldsymbol{P}_l$ is the projection matrix. Specifically, $\boldsymbol{P}_l$ is designed as projecting any vectors into the subspace orthogonal to the one spanned by all previously trained input vectors of W_l as follows:

$$\boldsymbol{P}_l = \boldsymbol{I} - \boldsymbol{A}_l (\boldsymbol{A}_l^T \boldsymbol{A}_l + \rho \boldsymbol{I})^{-1} \boldsymbol{A}_l^T \tag{25}$$

where $\boldsymbol{A}_l$ is the input matrix of the l^{th} layer and each column is the previously trained input vector of the weights $\boldsymbol{W}$, $\boldsymbol{I}$ is the unit matrix, and ρ is a small constant to make the $\boldsymbol{A}_l^T \boldsymbol{A}_l + \rho \boldsymbol{I}$ invertible.

According to the OWM method, the capacity of continuous learning of each layer is related to the rank of matrix $\boldsymbol{A}_l$. When the matrix $\boldsymbol{A}_l$ becomes full rank, the l^{th} layer runs out the capacity to learn new tasks. For the recurrent weights of RNN, the input matrix consists of previously trained time-varying firing rates. When the dimension of time is much larger than the dimension of neurons, the input matrix will become full rank rapidly. Therefore, constructing the input matrix with time-varying firing rates directly is not applicable. For the RNN with consistent population response, time-varying firing rates have certain pattern and are highly correlated among different time steps. Therefore, reducing the time dimension with PCA and constructing the input matrix with reduced firing rates is a possible solution to solve the problem.

For continuous learning of RNN in multiple tasks, the OWM method is improved as follows. First, the input matrix of recurrent weights $\boldsymbol{W}$ of RNN in the v^{th} task is constructed

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCDS.2020.3045574, IEEE Transactions on Cognitive and Developmental Systems

with all previously trained firing rates of hidden neurons as follows:

$$\boldsymbol{A}_H^v = [\boldsymbol{H}_1^v, ..., \boldsymbol{H}_K^v] \tag{26}$$

where $\boldsymbol{H}_i^v \in \mathbb{R}^{N\times T_t}$ records the time-varying firing rates of N hidden neurons during T_t time steps for the i^{th} movement targets after training in the v^{th} task and $\boldsymbol{A}_H^v \in \mathbb{R}^{N\times N_A}$ collects all firing rates for K trained movement targets in the v^{th} task, $N_A = K \times T_t$.

Then, the low-dimensional input matrix in the v^{th} task is computed through PCA without normalization as follows:

$$\tilde{\boldsymbol{A}}_H^v = \boldsymbol{A}_H^v \boldsymbol{Q}_v \tag{27}$$

where $\tilde{\boldsymbol{A}}_H^v \in \mathbb{R}^{N\times q}$ is the reduced matrix of $\boldsymbol{A}_H^v$. $\boldsymbol{Q}^v \in \mathbb{R}^{N_A\times q}$ is the transformed matrix. Each column of $\boldsymbol{Q}^v$ is an eigenvector of $(A_H^v)^T \boldsymbol{A}_H^v$ and $\boldsymbol{Q}_v$ selects q eigenvectors with maximal eigenvalues. It is noted that the $\tilde{\boldsymbol{A}}_H^v$ is reduced from the original input matrix $\boldsymbol{A}_H^v$ without normalization.

Correspondingly, the low-dimensional input matrices of all v learned tasks are collected in $\tilde{\boldsymbol{A}}_H$ as follows:

$$\tilde{\boldsymbol{A}}_H = [\tilde{\boldsymbol{A}}_H^1, ..., \tilde{\boldsymbol{A}}_H^v] \tag{28}$$

With the reduced input matrix, the orthogonal projection matrix of recurrent weights in the $(v+1)^{th}$ task is constructed as follows:

$$\begin{aligned} &\boldsymbol{P}_W = \boldsymbol{I} - \tilde{\boldsymbol{A}}_H(\tilde{\boldsymbol{A}}_H^T\tilde{\boldsymbol{A}}_H + \alpha_P \boldsymbol{I})^{-1}\tilde{\boldsymbol{A}}_H^T \\ &\boldsymbol{P}w\tilde{\boldsymbol{A}}_H = 0 \end{aligned} \tag{29}$$

where $\boldsymbol{P}_W \in \mathbb{R}^{N\times N}$ is the orthogonal projection matrix for $\boldsymbol{W}$ and orthogonal to input spaces of $\boldsymbol{W}$ in all learned tasks approximately.

Therefore, the increment of recurrent weights in a new task can be computed as follows:

$$\Delta\boldsymbol{W}^C = \Delta\boldsymbol{W}\boldsymbol{P}_W \tag{30}$$

where $\Delta\boldsymbol{W}^C$ is the increment of $\Delta\boldsymbol{W}$ after orthogonal projection, $\Delta\boldsymbol{W}$ is the weights adjustment computed based on reward modulated learning method in Section III-B.

As $\boldsymbol{P}_W\tilde{\boldsymbol{A}}_H = 0$, $\boldsymbol{P}_W\tilde{\boldsymbol{A}}_H^1 = ... = \boldsymbol{P}_W\tilde{\boldsymbol{A}}_H^v = 0$ holds. Therefore, the update of recurrent weights has little influence on previously learned knowledge as follows:

$$\begin{aligned} &(\boldsymbol{W} + \Delta\boldsymbol{W}^C)\boldsymbol{A}_H \\ &= \boldsymbol{W}\boldsymbol{A}_H + \Delta\boldsymbol{W}\boldsymbol{P}_W\boldsymbol{A}_H \\ &= \boldsymbol{W}\boldsymbol{A}_H + [\Delta\boldsymbol{W}\boldsymbol{P}_W\boldsymbol{A}_H^1, ..., \Delta\boldsymbol{W}\boldsymbol{P}_W\boldsymbol{A}_H^v] \\ &\approx \boldsymbol{W}\boldsymbol{A}_H + [\Delta\boldsymbol{W}\boldsymbol{P}w\tilde{\boldsymbol{A}}_H^1\boldsymbol{Q}_1^\dagger, ..., \Delta\boldsymbol{W}\boldsymbol{P}w\tilde{\boldsymbol{A}}_H^v\boldsymbol{Q}_v^\dagger,] \\ &= \boldsymbol{W}\boldsymbol{A}_H \end{aligned} \tag{31}$$

where $\boldsymbol{A}_H$ collects input matrices of all learned v tasks, $\boldsymbol{Q}_v^\dagger$ is the pseudo-inverse of $\boldsymbol{Q}_v$.

IV. Experiment

In this section, the effectiveness of proposed method is verified. In the experiments, the RNN is trained to control a sophisticated musculoskeletal system in multiple tasks. With the proposed method, the RNN can learn motor skills in multiple tasks without catastrophic forgetting. The RNN learned from training targets also demonstrates great generalization to unlearned targets. Furthermore, the RNN has robustness and can maintain the pattern of neuron activities under the perturbation of noises. In addition, the biological plausibility of the RNN is analyzed and verified in terms of neural population dynamics and muscle synergies.

Parameters	Symbol	Value
Number of input units	d	2
Number of hidden neurons	N	200
Number of output units	M	9
Time step	Δt	$1ms$
Decay of potential membranes	α	0.1
Probability of weights	p	0.7
Normal distribution of initial $\boldsymbol{W}$		$N(0, 0.01)$
Uniform distribution of initial $\boldsymbol{U}$		$U(-0.1, 0.1)$
Uniform distribution of initial $\mathbf{b}$		$U(-0.6, 0.6)$
Uniform distribution of initial $\boldsymbol{V}$		$U(-0.1, 0.1)$

TABLE I: Parameters of the RNN.

Parameters	Symbol	Value
Initial learning rate	γ_η	0.4
Initial filter factor	γ_{α^R}	0.3
Initial noise variance	γ_{σ^2}	2
Decay coefficient	τ_p	120
Maximum norm of gradients in 1^{st} task	g_1	0.001
Maximum norm of gradients in 2^{nd} task	g_2	0.005
Maximum norm of gradients in 3^{rd} task	g_3	0.005
Reduced dimensions	q	20
Factor of the cost function	ι_1	1
Factor of the cost function	ι_2	0.005
Factor of the cost function	ι_3	0.25

TABLE II: Learning parameters.

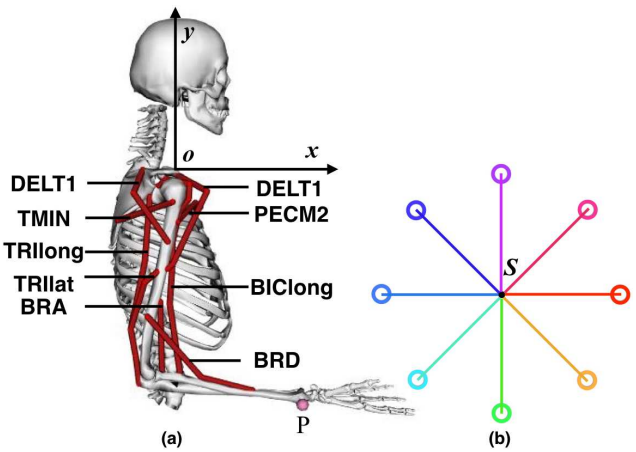

Fig. 3: Experimental setup. (a) shows the musculoskeletal system applied in the experiment and its muscle arrangment. (b) shows the isometric center-out reaching task. The musculoskeletal system is expected to move the end-effector P from the starting position S to the peripheral targets.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCDS.2020.3045574, IEEE Transactions on Cognitive and Developmental Systems

A. Experimental Setup

A musculoskeletal system with nine muscles and two degrees of freedom is applied in the experiments, which is shown in Fig. 3. The upper extremity of this musculoskeletal system can move in the sagittal plane. The degrees of freedom include shoulder flexion/extension and elbow flexion/extension. DELT1, DELT3, PECM2, TMIN, TRIlong, TRIlat, BIClong, BRA, and BRD are muscles utilized in the system with human-like muscular arrangements. The dynamics of muscle and musculoskeletal system are introduced in details in [44]. The establishment and modification of the musculoskeletal system are based on an open-source platform called OpenSim [61].

The center-out reaching task is applied in the experiments to verify the effectiveness of the proposed method, which is an ordinary paradigm in the research of motor neuroscience. As shown in the (b) of Fig. 3, the musculoskeletal system is expected to move the end-effector P from the starting position S to peripheral targets.

In order to realize the neuromuscular control of the musculoskeletal system, a RNN is designed to transform movement targets into muscle excitations. The structure and parameters of the RNN are listed in Table I. The two input units of the RNN receive the position coordinates of each movement target in x and y axes. The nine output units corresponds to the excitations of nine muscles. The duration of each movement is $0.4s$ and the time-step of the simulation is $1ms$. The weights are initialized in a range to guarantee efficient learning and motor-cortex-like neural population dynamics. The elements in weight matrices $\boldsymbol{W}, \boldsymbol{U}, \mathbf{b}, \mathbf{V}$ are initialized to zero with the probability of $1-p$. Non-zero elements of $\boldsymbol{W}$ are sampled from a normal distribution. As talked in Section III-A, the variance of the normal distribution is designed to guarantee both consistent population response and representation ability of RNN. For $\boldsymbol{U}$, $\mathbf{b}$, and $\boldsymbol{V}$, non-zero elements are initialized based on uniform distributions. During the training period, a cost function is designed to evaluate and update the performance of RNN as follows:

$$L = -R = \iota_1(\boldsymbol{p}-\boldsymbol{p}_d)^T(\boldsymbol{p}-\boldsymbol{p}_d) + \iota_2\dot{\boldsymbol{p}}^T\dot{\boldsymbol{p}} + \iota_3\left|\frac{\sum\lambda_j^+}{\sum\lambda_j^-}\right| \quad (32)$$

where $\boldsymbol{p}$, $\dot{\boldsymbol{p}}$, and $\boldsymbol{p}_d$ are the position, speed and desired position of the end-effector at the end of movement respectively. λ_j^+ and λ_j^- are positive and negative eigenvalues of the $(\boldsymbol{W}^+ - \boldsymbol{I})^T + (\boldsymbol{W}^+ - \boldsymbol{I})$ respectively. ι_1, ι_2, and ι_3 are factors to balance the weights of position error, speed error, and dynamic characteristics of RNN in the cost function. The setup of learning parameters is given in Table II.

B. Effectiveness of Continuous Motion Learning with Targets on Different Circles

In the section, three tasks are designed and movement targets are distributed on circles with the radii of $0.14m$, $0.10m$, and $0.05m$ respectively. Correspondingly, the RNN is trained by three times. In the first task, the RNN is trained to control the musculoskeletal system to reach targets on the circle with the radius of $0.14m$. The RNN is updated with the method described in Section III-B. In the second task, the RNN is trained again with new targets. The weights of RNN are updated on the basis of the ones learned from the first task and using the continuous learning method in Section III-C. In the third task, the RNN is trained again by the same way.

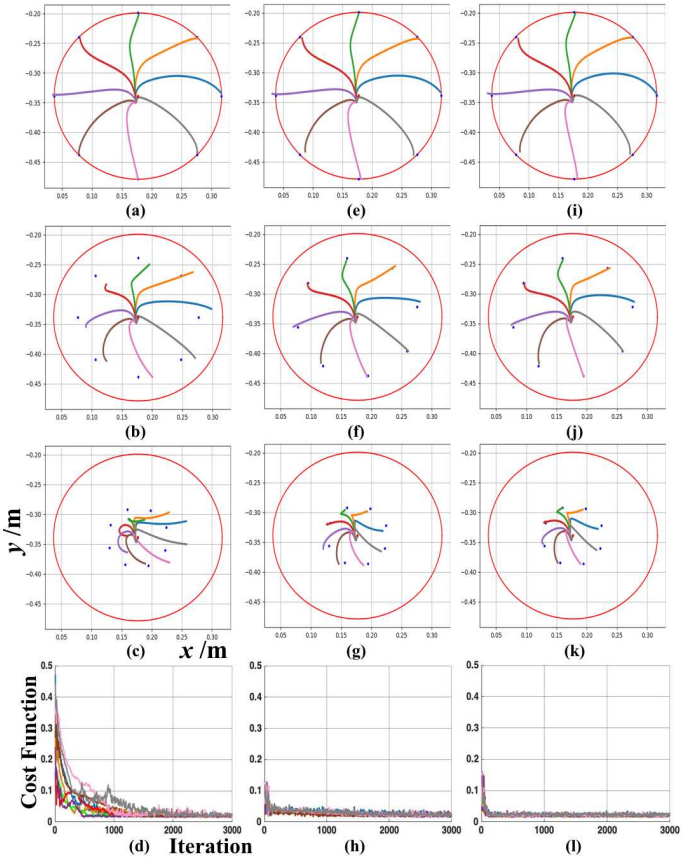

Fig. 4: Performance of continuous motion learning with targets on three different circles. (a)-(c), (e)-(g), and (i)-(k) show the performance in three tasks after the motion learning of RNN in the first, second, and third task respectively. The blue dots are movement targets in different tasks. (d), (h), and (l) are the cost functions during learning in the first, second, and third task respectively.

	1^{st} task (mm)	2^{nd} task (mm)	3^{rd} task (mm)
1^{st} learning	2.8 ± 1.3	22.2 ± 2.4	33.5 ± 2.3
2^{nd} learning	5.5 ± 2.4	4.9 ± 2.9	8.0 ± 2.3
3^{rd} learning	5.2 ± 2.3	3.8 ± 2.3	4.5 ± 2.3

TABLE III: Errors of training targets on three different circles after motion learning.

Based on the proposed method, the RNN can learn to transform movement targets into muscle excitations in the center-out reaching task. Furthermore, the motor skill in three tasks with different movement targets has been learned sequentially without catastrophic forgetting, which is shown in Fig. 4.

The performance of RNN is tested with all training targets in three tasks after each motion learning. Specifically, the (a)-(c) in Fig. 4 indicate that the RNN after the first motion learning can only control the musculoskeletal system to reach

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCDS.2020.3045574, IEEE Transactions on Cognitive and Developmental Systems

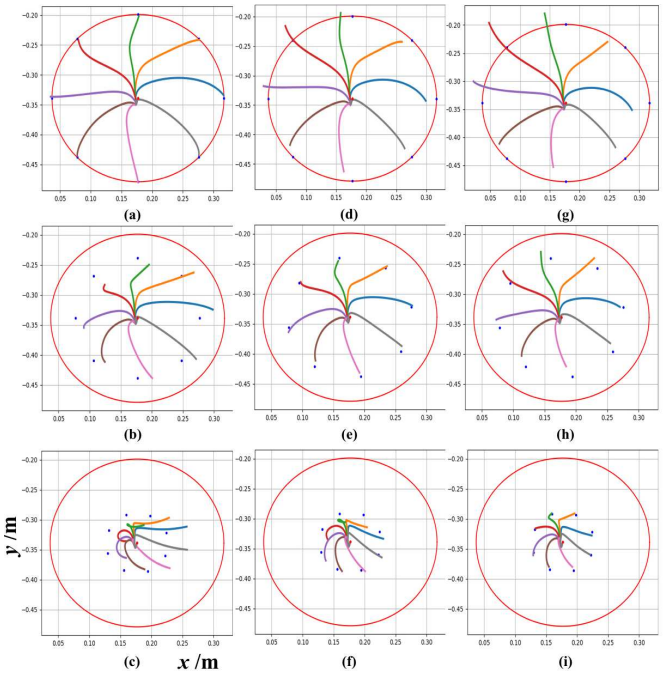

Fig. 5: Performance of motion learning without OWM in three tasks. (a)-(c), (d)-(f), and (g)-(i) show the performance in three tasks after the motion learning of RNN in the first, second, and third task respectively. The blue dots are movement targets in different tasks.

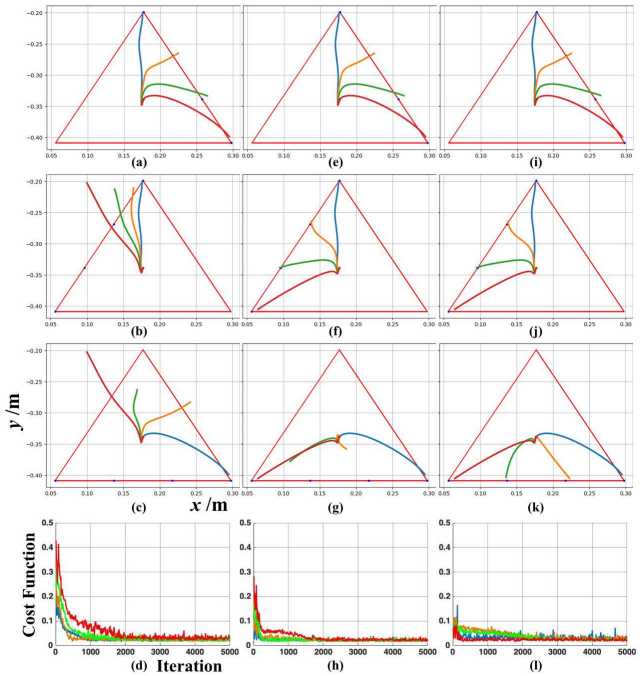

Fig. 6: Performance of continuous motion learning with targets on three different lines. (a)-(c), (e)-(g), and (i)-(k) show the performance in three tasks after the motion learning of RNN in the first, second, and third task respectively. The blue dots on each line are movement targets in each task. (d), (h), and (l) are the cost functions during learning in the first, second, and third task respectively.

training targets in the first task precisely. The (e)-(g) in Fig. 4 demonstrate that the RNN learns new targets in the second task with little influence on the motor skill learned in the first task. The (i)-(k) in Fig. 4 show that the RNN learns the motor skill in the third task and maintain the ones learned from the first and second task without catastrophic forgetting. Specifically, the training errors of targets on three different circles is shown in Table III. Therefore, the RNN can control the musculoskeletal system to reach all training targets in three tasks. With the observation of (d), (h), and (l) in Fig. 4, the cost functions in new tasks decrease more rapidly and the learned motor skills in previous tasks accelerate the learning in new tasks.

In contrast, the motion learning is also performed without the improved OWM method, which is shown in Fig. 5. The RNN can learn the motor skill in three tasks individually. However, the previously learned knowledge will be forgotten rapidly, which also proves the effectiveness of the proposed method on continuous learning from the reverse side.

C. Effectiveness of Continuous Motion Learning with Targets on Different Lines

In the section, three tasks are designed and movement targets are distributed on three different lines. Correspondingly, the RNN is trained by three times with the same way described in Section IV-B. Comparing with the distribution of targets on different circles, the difference of targets on three lines is enhanced. As shown in the Fig. 6, the RNN can also learn the motor skill in three tasks individually without catastrophic forgetting, which further verifies the effectiveness of the proposed method.

D. Motion Generalization

Furthermore, the learned motor skill of RNN can be not only applied to training targets but also generalized to unlearned targets. As shown in Fig. 7, the performance of RNN is tested at 80 unlearned targets. Based on the motor skill learned from the three tasks, movements from the origin to peripheral targets within the circle with the radius of $0.14m$ can all be realized. The average errors during movements in (a), (b), (c), and (d) are $6.4\pm4.2\ mm$, $4\pm1.9\ mm$, $4.5\pm2.1\ mm$, and $3.9\pm1.6\ mm$ respectively.

E. Robustness for Noises

Benefiting from the relative smooth dynamics and consistent population response, the RNN also demonstrates great robustness for noises. As shown in Fig. 8, normal noises with different magnitudes are applied to neuron activities for a period of time. After the noises disappear, the firing rate of each hidden neuron returns to the neighborhood of its expected state without perturbation. Furthermore, the change rate of firing rate also converges to zero approximately. Although the movements controlled by the perturbed RNN deviate from the expected trajectories, the dynamic system of RNN is still stable and the tendencies of moving to different targets still exist.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCDS.2020.3045574, IEEE Transactions on Cognitive and Developmental Systems

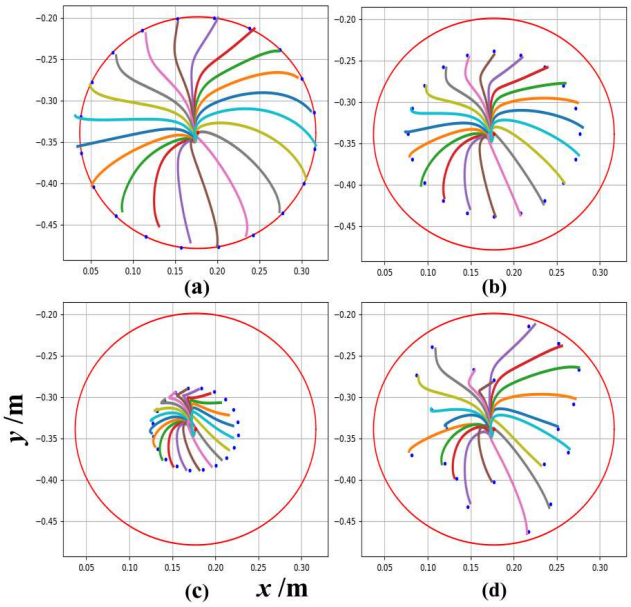

Fig. 7: Performance of motion generalization on unlearned movement targets. (a), (b), (c), and (d) demonstrate the performance at 20 unlearned targets respectively. The targets in (a), (b), and (c) are distributed on the circle with the radius of 0.14m, 0.10m, and 0.05m. The targets in (d) is randomly selected within the circle with the radius of 0.14m.

F. Biological Plausibility

In the experiments, the proposed method also demonstrates great biological plausibility. The activities of neurons and muscles after motion learning resemble the observations of the monkey and human. Specifically, the firing rates of hidden neurons in the RNN demonstrate motor-cortex-like consistent population response during different movements. Furthermore, the muscle excitations of different movements show human-like muscle synergies.

In the experiments, the population activity of RNN is analyzed. The firing rates of all hidden neurons can be regarded as the population activity. As shown in the (a)-(d) of Fig. 9, the firing rates of neurons during different movements are different but the change rates of most neuronal firing rates converge to zero uniformly. The firing rates of all neurons is a high dimensional state trajectory and can be projected into a two-dimensional state space through the PCA method. As shown in the (e) of Fig. 9, the low-dimensional population activities during different movements are consistent and demonstrate orderly rotation structure like the observation of motor cortex.

Furthermore, the pattern of muscle excitations is also analyzed. As shown in Fig. 10, (a) and (b) demonstrate the modulation of muscle excitations during movements with the same speed and different directions respectively. With the comparison of (a) and (b), the modulation of muscle excitations with different movement speeds is also demonstrated, which can be observed more clearly in (c) and (d). Based on above observations, muscle excitations of different movements are constructed by similar patterns and are modulated in terms of the movement directions and speeds.

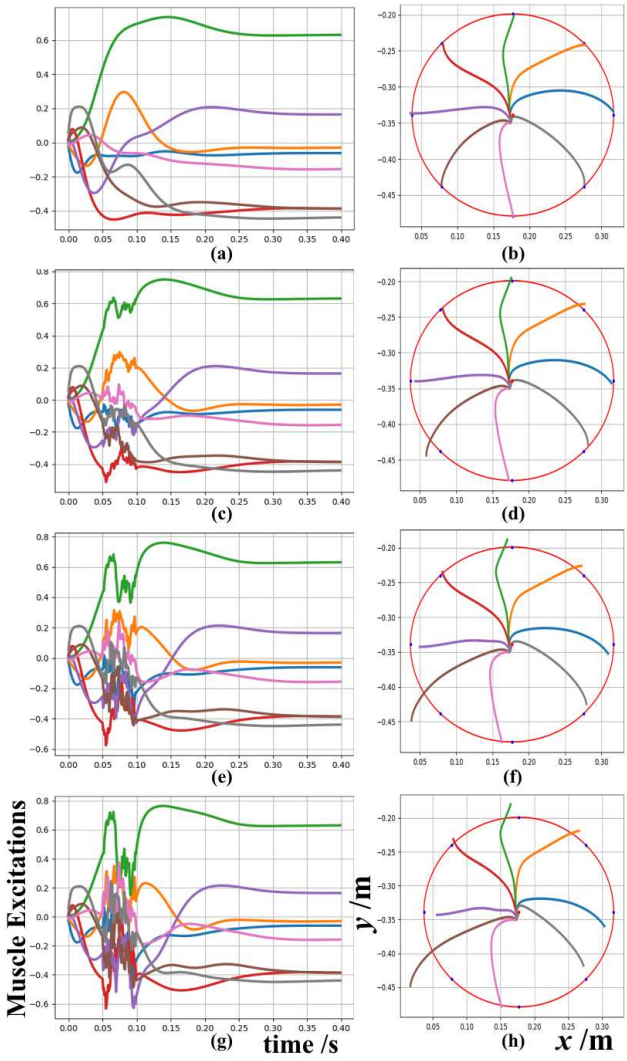

Fig. 8: Effectiveness of robustness for noises. (a) demonstrates the firing rates of 8 hidden neurons of all 200 ones without noises. (c), (e), and (g) show the situations under normal noises with the standard deviation of 0.025, 0.05, and 0.75 from 0.05s to 0.1s. (b), (d), (f), and (h) are movements of a musculoskeletal system controlled by the RNN under the situations of (a), (c), (e), and (g) respectively.

In order to extract the common patterns, muscle excitations are decomposed with the Non-negative Matrix Factorization method. Then, three muscle synergies are extracted from the muscle excitations of 20 different movements and these synergies can reconstruct the muscle excitations effectively with variance account for (VAF) of 93.85%, which is shown in the (a)-(c) of Fig. 11. As shown in the (d), the amplitudes of three synergies are modulated with the movement directions, which can explain the modulation of muscle excitations with movement directions.

V. Discussion

In this section, the comparison with some relevant work [39, 42, 43, 55] and the improvement in the future are discussed.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCDS.2020.3045574, IEEE Transactions on Cognitive and Developmental Systems

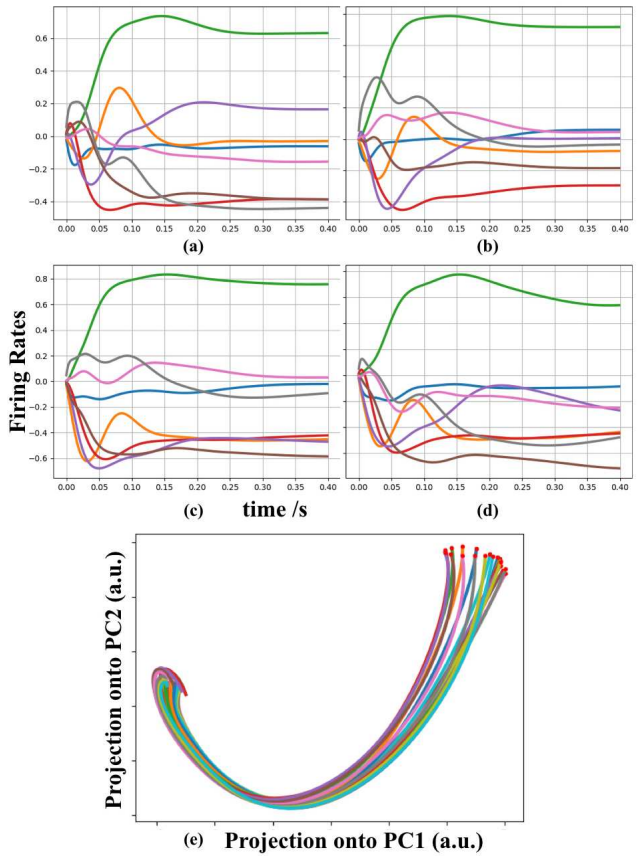

Fig. 9: (a), (b), (c), and (d) show the time-varying activities (firing rates) of 8 hidden neurons of all 200 ones during four movements from an origin to targets on a circle with radius of 0.14m. The targets are distributed on the direction of $0°$, $90°$, $180°$, and $270°$ respectively. (e) shows the projection of population response of hidden neurons into two-dimensional PCA space. Each curve corresponds to a movement on the direction from $0° - 360°$. Two PC dimensions are plotted versus each other.

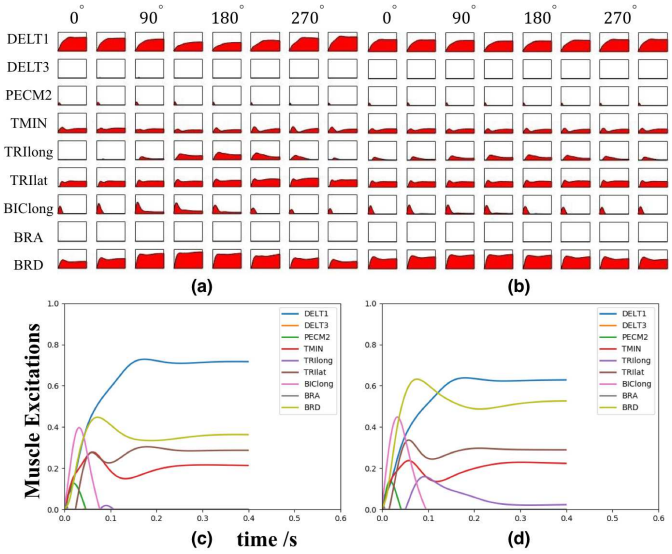

Fig. 10: Modulation of muscle excitations. (a) and (b) show muscle excitations during eight movements from an origin to targets on the circles with the radius of of 0.14m and 0.05m respectively. As the time duration of movements is the same, movements with different distances can be regarded as movements with different speeds. The targets are distributed at various directions from $0°$ to $360°$ with the interval of $45°$. (c) and (d) show muscle excitations during movements from an origin to targets at the same direction and different distances of 0.14m and 0.05m respectively.

The work in [55] trains the RNN to produces muscle activities with supervised learning and makes the RNN generate consistent population response with regularization. The work in [39] trains the RNN using reward-based learning with emotion modulation. The work in [42, 43] realizes the control of musculoskeletal system with establishing explicit muscle synergy models. In this paper, we unifies the muscle-synergies-based and RNN-based neuromuscular control through realizing muscle synergies using the RNN with consistent population response. The RNN is trained with reward modulated learning and the condition of consistent population response is investigated with Lyapunov analysis. Besides, the methods proposed in [39, 42, 43, 55] only realize motion learning in single movement task. However, the method in this paper has continuous learning ability. As shown in the Fig. 12 and Table IV, three different control methods are compared and the proposed method in this paper improves the precision of motion learning and generalization. Furthermore, the continuous learning ability in multiple tasks is also shown in the Fig. 4 and Fig. 6. In addition, the investigation of neuromuscular control is not only important for musculoskeletal robots and may also benefit the control of other non-linear, coupling robotic systems [62, 63].

In this paper, the states of neurons in the RNN and the muscle excitations are initialized with zeros for each movement. Therefore, muscles have not been activated to resist the effect of gravity at the beginning and the movement of musculoskeletal system will be affected under this condition. For human and animals, the premotor cortex and primary motor cortex will prepare the movements and generate an appropriate initialization for movement execution [64, 65]. Consequently, they can hold on the initial posture and resist gravity with appropriate muscle excitations. In the future, we will improve the method with appropriate motor-cortex-inspired movement preparation. Furthermore, we will also gradually apply our neuromuscular control method to the hardware of musculoskeletal robots and promotes the development of the musculoskeletal robot. The reward modulated learning may be further improved with evolutionary algorithms and other more complicated reinforcement architectures [66–68]. In addition, the musculoskeletal system with flexibility and compliance may also be applied to realize human-like manipulation and collaboration task [69–71].

VI. Conclusion

Inspired by how human control the musculoskeletal system, a novel neuromuscular control method is proposed in this

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCDS.2020.3045574, IEEE Transactions on Cognitive and Developmental Systems

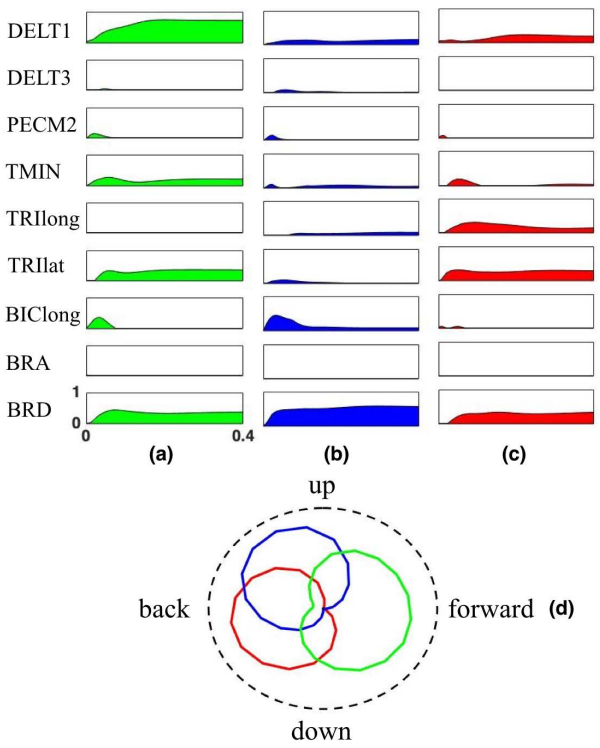

Fig. 11: Extracted muscle synergies and their modulation. (a), (b), and (c) show muscle synergies extracted from muscle excitations during twenty movements with various directions. (d) shows the modulation of amplitudes of three muscle synergies with the movement directions.

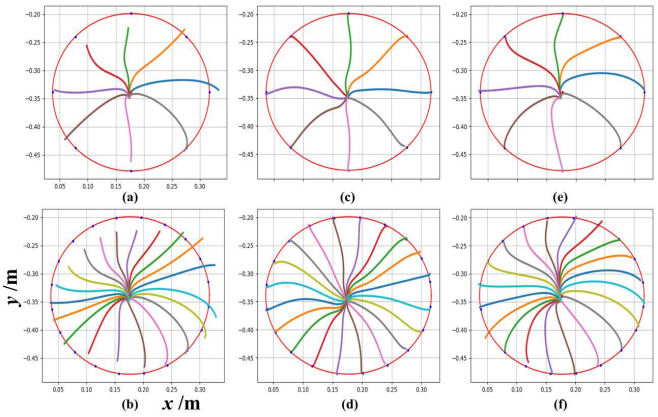

Fig. 12: Comparison results of three different control methods of a musculoskeletal system. (a), (c) and (e) show the performance of the method in [39], [43], and this paper on 8 training targets respectively. (b), (d) and (f) show the performance of the method in [39], [43], and this paper on 20 unlearned targets respectively.

paper. This method improves both of the performance of neuromuscular control and biological plausibility. With the continuous reward modulated learning, the learning ability is enhanced and multiple tasks can be learned. With the consistent population response of the RNN, great generalization and robustness for noises are achieved. Furthermore, the method also demonstrates the motor-like consistent population response and muscle-like synergy, which proves the biological plausibility and validate the effectiveness of neural mechanisms to some extent.

	Training Errors (mm)	Generalization Errors (mm)
Method in [39]	16.9 ± 7.9	19.7 ± 8.5
Method in [43]	5.1 ± 1.7	5.6 ± 2.0
Method in this paper	2.8 ± 1.3	5.0 ± 2.7

TABLE IV: Comparison of training errors and generalization errors of three different methods

REFERENCES

[1] I. Mizuuchi, R. Tajima, T. Yoshikai, D. Sato, K. Nagashima, M. Inaba, Y. Kuniyoshi, and H. Inoue, "The design and control of the flexible spine of a fully tendon-driven humanoid" kenta"," in *IEEE/RSJ international conference on intelligent robots and systems*, vol. 3. IEEE, 2002, pp. 2527–2532.

[2] I. Mizuuchi, Y. Nakanishi, Y. Sodeyama, Y. Namiki, T. Nishino, N. Muramatsu, J. Urata, K. Hongo, T. Yoshikai, and M. Inaba, "An advanced musculoskeletal humanoid kojiro," in *2007 7th IEEE-RAS International Conference on Humanoid Robots*. IEEE, 2007, pp. 294–299.

[3] O. Holland and R. Knight, "The anthropomimetic principle," in *Proceedings of the AISB06 symposium on biologically inspired robotics*, 2006, pp. 1–8.

[4] I. Mizuuchi, T. Yoshikai, Y. Sodeyama, Y. Nakanishi, A. Miyadera, T. Yamamoto, T. Niemela, M. Hayashi, J. Urata, Y. Namiki *et al.*, "Development of musculoskeletal humanoid kotaro," in *Proceedings 2006 IEEE International Conference on Robotics and Automation*. IEEE, 2006, pp. 82–87.

[5] Y. Nakanishi, Y. Asano, T. Kozuki, H. Mizoguchi, Y. Motegi, M. Osada, T. Shirai, J. Urata, K. Okada, and M. Inaba, "Design concept of detail musculoskeletal humanoid kenshiro-toward a real human body musculoskeletal simulator," in *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*. IEEE, 2012, pp. 1–6.

[6] M. Jäntsch, S. Wittmeier, K. Dalamagkidis, A. Panos, F. Volkart, and A. Knoll, "Anthrob-a printed anthropomimetic robot," in *2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*. IEEE, 2013, pp. 342–347.

[7] C. Richter, S. Jentzsch, R. Hostettler, J. A. Garrido, E. Ros, A. C. Knoll, F. Röhrbein, P. van der Smagt, and J. Conradt, "Scalability in neural control of musculoskeletal robots," *arXiv preprint arXiv:1601.04862*, 2016.

[8] Y. Asano, K. Okada, and M. Inaba, "Design principles of a human mimetic humanoid: Humanoid platform to study human intelligence and internal body system," *Science Robotics*, vol. 2, no. 13, p. eaaq0899, 2017.

[9] K. Narioka, R. Niiyama, Y. Ishii, and K. Hosoda, "Pneumatic musculoskeletal infant robots," in *Proceedings of the 2009 IEEE/RSJ International conference on intelligent robots and systems*, 2009.

[10] K. Narioka, T. Homma, and K. Hosoda, "Humanlike ankle-foot complex for a biped robot," in *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*. IEEE, 2012, pp. 15–20.

[11] I. Mizuuchi, M. Kawamura, T. Asaoka, and S. Kumakura, "Design and development of a compressor-embedded pneumatic-driven musculoskeletal humanoid," in *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*. IEEE, 2012, pp. 811–816.

[12] S. Ikemoto, F. Kannou, and K. Hosoda, "Humanlike shoulder complex for musculoskeletal robot arms," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 4892–4897.

[13] K. Ogawa, K. Narioka, and K. Hosoda, "Development of whole-body humanoid pneumat-bs with pneumatic musculoskeletal system," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2011, pp. 4838–4843.

[14] H. Shin, S. Ikemoto, and K. Hosoda, "Understanding function of gluteus medius in human walking from constructivist approach," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 3894–3899.

[15] R. Niiyama and Y. Kuniyoshi, "A pneumatic biped with an artificial musculoskeletal system," in *Proceedings of 4th International Symposium on Adaptive Motion of Animals and Machines*, 2008, pp. 80–81.

[16] S. Kurumaya, K. Suzumori, H. Nabae, and S. Wakimoto, "Musculoskeletal lower-limb robot driven by multifilament muscles," *Robomech Journal*, vol. 3, no. 1, p. 18, 2016.

[17] S. Zhong, J. Chen, X. Niu, H. Fu, and H. Qiao, "Reducing redundancy of musculoskeletal robot with convex hull vertexes selection," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 12, no. 3, pp. 601–617, 2020.

[18] M. C. Yip and G. Niemeyer, "High-performance robotic muscles from conductive nylon sewing thread," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 2313–2318.

[19] H.-I. Kim, M.-W. Han, W. Wang, S.-H. Song, H. Rodrigue, and S.-H. Ahn, "Design and development of bio-mimetic soft robotic hand with shape memory alloy," in *2015 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2015, pp. 2330–2334.

[20] A. Miriyev, K. Stack, and H. Lipson, "Soft material for soft actuators," *Nature communications*, vol. 8, no. 1, pp. 1–8, 2017.

[21] Y. Morimoto, H. Onoe, and S. Takeuchi, "Biohybrid robot powered by an antagonistic pair of skeletal muscle tissues," *Science Robotics*, vol. 3, no. 18, p. eaat4440, 2018.

[22] R. Niiyama, S. Nishikawa, and Y. Kuniyoshi, "Athlete robot with applied human muscle activation patterns for bipedal running," in *2010 10th IEEE-RAS International Conference on Humanoid Robots*. IEEE, 2010, pp. 498–503.

[23] M. Jantsch, C. Schmaler, S. Wittmeier, K. Dalamagkidis, and A. Knoll, "A scalable joint-space controller for musculoskeletal robots with spherical joints," in *2011 IEEE International Conference on Robotics and Biomimetics*. IEEE, 2011, pp. 2211–2216.

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

[25] S. Ookubo, Y. Asano, T. Kozuki, T. Shirai, K. Okada, and M. Inaba, "Learning nonlinear muscle-joint state mapping toward geometric model-free tendon driven musculoskeletal robots," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2015, pp. 765–770.

[26] M. Kawamura, S. Ookubo, Y. Asano, T. Kozuki, K. Okada, and M. Inaba, "A joint-space controller based on redundant muscle tension for multiple dof joints in musculoskeletal humanoids," in *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2016, pp. 814–819.

[27] K. Kawaharazuka, M. Kawamura, S. Makino, Y. Asano, K. Okada, and M. Inaba, "Antagonist inhibition control in redundant tendon-driven structures based on human reciprocal innervation for wide range limb motion of musculoskeletal humanoids," *IEEE Robotics and Automation Letters*, vol. 2, no. 4, pp. 2119–2126, 2017.

[28] K. Tahara, S. Arimoto, M. Sekimoto, and Z.-W. Luo, "On control of reaching movements for musculo-skeletal redundant arm model," *Applied Bionics and Biomechanics*, vol. 6, no. 1, pp. 11–26, may 2009.

[29] K. Tahara, Y. Kuboyama, and R. Kurazume, "Iterative learning control for a musculoskeletal arm: Utilizing multiple space variables to improve the robustness," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, oct 2012, pp. 4620–4625.

[30] H. Dong, N. Figueroa, and A. E. Saddik, "Muscle force control of a kinematically redundant bionic arm with real-time parameter update," in *2013 IEEE International Conference on Systems, Man, and Cybernetics*. IEEE, oct 2013, pp. 1640–1647.

[31] M. H. B. E., R. Vatankhah, M. Broushaki, and A. Alasty, "Adaptive optimal multi-critic based neuro-fuzzy control of MIMO human musculoskeletal arm model," *Neurocomputing*, vol. 173, pp. 1529–1537, jan 2016.

[32] D. G. Thelen, F. C. Anderson, and S. L. Delp, "Generating dynamic simulations of movement using computed muscle control," *Journal of Biomechanics*, vol. 36, no. 3, pp. 321–328, mar 2003.

[33] D. Stanev and K. Moustakas, "Simulation of constrained musculoskeletal systems in task space," *IEEE Transactions on Biomedical Engineering*, vol. 65, no. 2, pp. 307–318, 2017.

[34] L. Jin, Z. Xie, M. Liu, C. Ke, C. Li, and C. Yang, "Novel joint-drift-free scheme at acceleration level for robotic redundancy resolution with tracking error theoretically eliminated," *IEEE/ASME Transactions on Mechatronics*, 2020.

[35] N. Khan and I. Stavness, "Prediction of muscle activations for reaching movements using deep neural networks," *arXiv preprint arXiv:1706.04145*, 2017.

[36] M. Nakada, T. Zhou, H. Chen, T. Weiss, and D. Terzopoulos, "Deep learning of biomimetic sensorimotor control for biomechanical human animation," *ACM Transactions on Graphics*, vol. 37, no. 4, pp. 1–15, jul 2018.

[37] L. Kidzi'nski, S. P. Mohanty, C. F. Ong, J. L. Hicks, S. F. Carroll, S. Levine, M. Salathé, and S. L. Delp, "Learning to run challenge: Synthesizing physiologically accurate motion using deep reinforcement learning," in *The NIPS'17 Competition: Building Intelligent Systems*. Springer, 2018, pp. 101–120.

[38] L. Kidzi'nski, S. P. Mohanty, C. F. Ong, Z. Huang, S. Zhou, A. Pechenko, A. Stelmaszczyk, P. Jarosik, M. Pavlov, S. Kolesnikov *et al.*, "Learning to run challenge solutions: Adapting reinforcement learning methods for neuromusculoskeletal environments," in *The NIPS'17 Competition: Building Intelligent Systems*. Springer, 2018, pp. 121–153.

[39] X. Huang, W. Wu, H. Qiao, and Y. Ji, "Brain-inspired motion learning in recurrent neural network with emotion modulation," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 10, no. 4, pp. 1153–1164, dec 2018.

[40] J. Zhou, J. Chen, H. Deng, and H. Qiao, "From rough to precise: A human-inspired phased target learning framework for redundant musculoskeletal systems," *Frontiers in neurorobotics*, vol. 13, p. 61, 2019.

[41] L. Kidzi'nski, C. Ong, S. P. Mohanty, J. Hicks, S. Carroll, B. Zhou, H. Zeng, F. Wang, R. Lian, H. Tian *et al.*, "Artificial intelligence for prosthetics: Challenge solutions," in *The NeurIPS'18 Competition*. Springer, 2020, pp. 69–128.

[42] E. Rckert and A. d'Avella, "Learned parametrized dynamic movement primitives with shared synergies for controlling robotic and musculoskeletal systems," *Frontiers in Computational Neuroscience*, vol. 7, p. 138, 2013.

[43] J. Chen, S. Zhong, E. Kang, and H. Qiao, "Realizing human-like manipulation with a musculoskeletal system and biologically inspired control scheme," *Neurocomputing*, vol. 339, pp. 116–129, apr 2019.

[44] J. Chen and H. Qiao, "Muscle-synergies-based neuromuscular control for motion learning and generalization of a musculoskeletal system," *IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2020*, 2020.

[45] J. Li, Z. Li, X. Li, Y. Feng, Y. Hu, and B. Xu, "Skill learning strategy based on dynamic motion primitives for human-robot cooperative manipulation," *IEEE Transactions on Cognitive and Developmental Systems*, 2020.

[46] A. d'Avella, P. Saltiel, and E. Bizzi, "Combinations of muscle synergies in the construction of a natural motor behavior," *Nature neuroscience*, vol. 6, no. 3, pp. 300–308, 2003.

[47] A. d'Avella, L. Fernandez, A. Portone, and F. Lacquaniti, "Modulation of phasic and tonic muscle synergies with reaching direction and speed," *Journal of neurophysiology*, vol. 100, no. 3, pp. 1433–1454, 2008.

[48] S. A. Overduin, A. d'Avella, J. Roh, J. M. Carmena, and E. Bizzi, "Representation of muscle synergies in the primate brain," *Journal of Neuroscience*, vol. 35, no. 37, pp. 12 615–12 624, 2015.

[49] E. V. Evarts, "Relation of pyramidal tract activity to force exerted during voluntary movement." *Journal of neurophysiology*, vol. 31, no. 1, pp. 14–27, 1968.

[50] R. P. Dum and P. L. Strick, "The origin of corticospinal projections from the premotor areas in the frontal lobe," *Journal of Neuroscience*, vol. 11, no. 3, pp. 667–689, 1991.

[51] A. P. Georgopoulos, A. B. Schwartz, and R. E. Kettner, "Neuronal population coding of movement direction," *Science*, vol. 233, no. 4771, pp. 1416–1419, 1986.

[52] A. B. Schwartz, "Direct cortical representation of drawing," *Science*, vol. 265, no. 5171, pp. 540–542, 1994.

[53] D. W. Moran and A. B. Schwartz, "Motor cortical representation of speed and direction during reaching," *Journal of neurophysiology*, vol. 82, no. 5, pp. 2676–2692, 1999.

[54] M. M. Churchland, J. P. Cunningham, M. T. Kaufman, J. D. Foster, P. Nuyujukian, S. I. Ryu, and K. V. Shenoy, "Neural population dynamics during reaching," *Nature*, vol. 487, no. 7405, pp. 51–56, 2012.

[55] D. Sussillo, M. M. Churchland, M. T. Kaufman, and K. V. Shenoy, "A neural network that finds a naturalistic solution for the production of muscle activity," *Nature neuroscience*, vol. 18, no. 7, pp. 1025–1033, 2015.

[56] A. A. Russo, S. R. Bittner, S. M. Perkins, J. S. Seely, B. M. London, A. H. Lara, A. Miri, N. J. Marshall, A. Kohn, T. M. Jessell *et al.*, "Motor cortex embeds muscle-like commands in an untangled population response," *Neuron*, vol. 97, no. 4, pp. 953–966, 2018.

[57] P. Zhang, J. Huang, W. Li, X. Ma, P. Yang, J. Dai, and J. He, "Using high-frequency local field potentials from multicortex to decode reaching and grasping movements in monkey," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 11, no. 2, pp. 270–280, 2018.

[58] D. G. Thelen, "Adjustment of muscle mechanics model parameters to simulate dynamic contractions in older adults," *Journal of Biomechanical Engineering*, vol. 125, no. 1, p. 70, 2003.
[59] M. Millard, T. Uchida, A. Seth, and S. L. Delp, "Flexing computational muscle: modeling and simulation of musculotendon dynamics," *Journal of biomechanical engineering*, vol. 135, no. 2, p. 021005, 2013.
[60] G. Zeng, Y. Chen, B. Cui, and S. Yu, "Continual learning of context-dependent processing in neural networks," *Nature Machine Intelligence*, vol. 1, no. 8, pp. 364–372, 2019.
[61] S. L. Delp, F. C. Anderson, A. S. Arnold, P. Loan, A. Habib, C. T. John, E. Guendelman, and D. G. Thelen, "OpenSim: Open-source software to create and analyze dynamic simulations of movement," *IEEE Transactions on Biomedical Engineering*, vol. 54, no. 11, pp. 1940–1950, nov 2007.
[62] J. Wang, Z. Wu, M. Tan, and J. Yu, "3-d path planning with multiple motions for a gliding robotic dolphin," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2019.
[63] ——, "Model predictive control-based depth control in gliding motion of a gliding robotic dolphin," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2019.
[64] M. T. Kaufman, M. M. Churchland, S. I. Ryu, and K. V. Shenoy, "Cortical activity in the null space: permitting preparation without movement," *Nature neuroscience*, vol. 17, no. 3, pp. 440–448, 2014.
[65] G. F. Elsayed, A. H. Lara, M. T. Kaufman, M. M. Churchland, and J. P. Cunningham, "Reorganization between preparatory and movement population responses in motor cortex," *Nature communications*, vol. 7, no. 1, pp. 1–15, 2016.
[66] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, "A reference vector guided evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 5, pp. 773–791, 2016.
[67] Y. Tian, R. Cheng, X. Zhang, and Y. Jin, "Platemo: A matlab platform for evolutionary multi-objective optimization [educational forum]," *IEEE Computational Intelligence Magazine*, vol. 12, no. 4, pp. 73–87, 2017.
[68] X. Huang, W. Wu, and H. Qiao, "Computational modeling of emotion-motivated decisions for continuous control of mobile robots," *IEEE Transactions on Cognitive and Developmental Systems*, 2020.
[69] X. Yu, S. Zhang, L. Sun, Y. Wang, C. Xue, and B. Li, "Cooperative control of dual-arm robots in different human-robot collaborative tasks," *Assembly Automation*, 2019.
[70] X. Li, Y. Qian, R. Li, X. Niu, and H. Qiao, "Robust form-closure grasp planning for 4-pin gripper using learning-based attractive region in environment," *Neurocomputing*, vol. 384, pp. 268–281, 2020.
[71] G. Wang, X. Hua, J. Xu, L. Song, and K. Chen, "A deep learning based automatic surface segmentation algorithm for painting large-size aircraft with 6-dof robot," *Assembly Automation*, 2019.

Jiahao Chen received the B.Eng. degree in Automation from China Agricultural University, Beijing, China, in 2016. He is currently a Ph.D. degree candidate at Institute of Automation, Chinese Academy of Sciences (CASIA), Beijing, China.

His current research interests include brain-inspired motion learning and multi-task learning of intelligent robots.

Hong Qiao (SM'06-F'18) received the B.Eng. degree in hydraulics and control and the M.Eng. degree in robotics from Xi'an Jiaotong University, Xi'an, China, in 1986 and 1989, respectively, the M.Phil. degree in robotics control from the Industrial Control Center, University of Strathclyde, Strathclyde, U.K., in 1992, and the Ph.D. degree in robotics and artificial intelligence from De Montfort University, Leicester, U.K., in 1995.

She was a University Research Fellow with De Montfort University from 1995 to 1997. She was a Research Assistant Professor with the Department of Manufacturing Engineering and Engineering Management, City University of Hong Kong, Hong Kong, from 1997 to 2000, where she was an Assistant Professor from 2000 to 2002. Since 2002, she has been a Lecturer with the School of Informatics, University of Manchester, Manchester, U.K. She is currently a Professor with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China. She first proposed the concept of the attractive region in strategy investigation, which has successfully been applied by herself in robot assembly, robot grasping, and part recognition. She has authored the book entitled Advanced Manufacturing Alert (Wiley, 1999). Her current research interests include information-based strategy investigation, robotics and intelligent agents, animation, machine learning, and pattern recognition.

Prof. Qiao is currently an Associate Editor of the IEEE TRANSACTIONS ON CYBERNETICS and the IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING. She is the Editor-in-Chief of the Assembly Automation. She is currently a member of the Administrative Committee of the IEEE Robotics and Automation Society, the IEEE Medal for Environmental and Safety Technologies Committee, the Early Career Award Nomination Committee, the Most Active Technical Committee Award Nomination Committee, and the Industrial Activities Board for RAS.