Biomimetic Force and Impedance Adaptation based on Broad Learning System in Stable and Unstable Tasks

Zhenyu Lu, *Member, IEEE* and Ning Wang, *Member, IEEE*

***Abstract*—This article presents a novel biomimetic force and impedance adaption framework based on Broad Learning System (BLS) for robot control in stable and unstable environments. Different from iterative learning control, the adaptation process is realized by a neural network (NN)-based framework, similar to BLS, to realize a varying learning rate for the feedforward force and impedance factors. The connections of NN layers and the settings of the feature nodes are related to human motor control and learning principle that is described as a relationship between feedforward force, impedance, reflex and position errors, etc., to make the NN explainable. Some comparative simulations are created and tested in five force fields to verify the advantages of the proposed framework in terms of force and trajectory tracking efficiency and accuracy, robust responses to different force situations and continuity of force application in a mixed stable and unstable environment. Finally, an experiment is taken to verify the effectiveness of the proposed method.**

***Index Terms*—Impedance, Feedforward force, Broad Learning System, Robotics control, Skill learning and generalization**

I. Introduction

VARIABLE force and impedance control mechanism is widely existed in human daily movements and work, such as drilling and grasping, requiring central nervous system (CNS) to control muscle reflexes and learning processes against intrinsic instability caused by unknown external forces, motor noises and raw material properties when interacting with objects in our environment. In recent years, research on motor skills learning has been used for robotic applications, such as teleoperation [1], transferring skills from humans to robots [2], and haptic interaction [3], [4], integrating the electromyography perception and learning technology.

To realize human motor learning process, a mechanism used by the CNS is proposed, involving both feedback and feedback control, to represent the relationship between motor commands and movement, so that the CNS can adapt the dynamics of the limb to compensate for mechanical instability and interaction forces from the environment [6]. However, even based on the common mechanism, there are some models describing human motor control and learning, but they are slightly different from each other. Franklin *et al.* built the model in which both inverse dynamics control and impedance control are active during the motor learning process to counteract mechanical instability rather than just predicting final learning outcomes [5]. In [6], a V-shaped

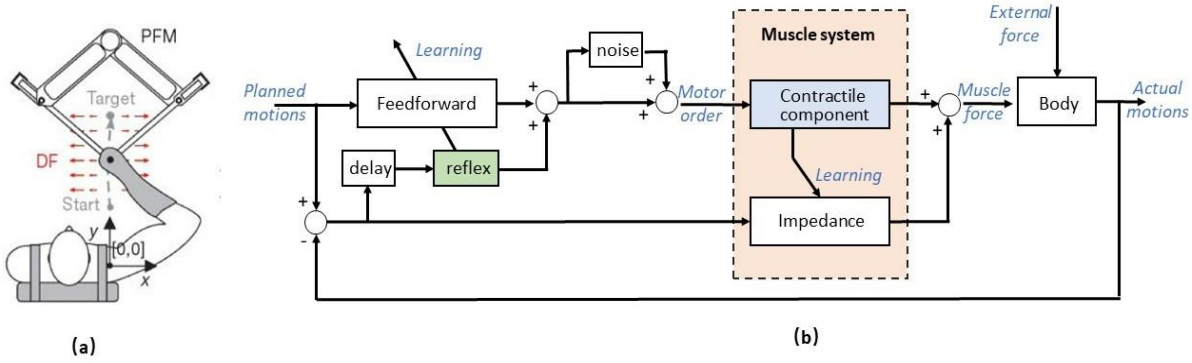

Fig. 1. Simulation diagram of human movements to investigate motor control and learning (a) Point-to-point movements with lateral instability produced by a parallel-link direct drive air-magnet floating manipulandum (PFM) to show human feedforward force and impedance adaptation [12]. (b) Control diagram of the controller with neural control and feedback error learning in [8].

This work was partially supported by the H2020 Marie Skłodowska-Curie Actions Individual Fellowship under Grant 101030691. (Corresponding author: *Ning Wang, E-mail: Ning2.Wang@uwe.ac.uk*).

The authors are with the Bristol Robotics Laboratory, University of the West of England, BS16 1QY, United Kingdom,

learning function is created to specify exactly how feedforward commands are adjusted to individual muscles based on tracking error. Tee *et al.* claimed the shortcomings of the above models and defined a model specifying the activation of each muscle that is adapted from one movement to the next and explains how humans learn to perform movements in the environment with novel dynamics of tool use [7].

In comparison, the model shown in [8] is based on a stability measure corresponding to Lyapunov exponents, which explores the learning mechanism of human arms for unstable interaction, and each muscle is not modeled separately as in [7]. As Fig. 1 shows, this diagram includes feedforward forces, motor noise, delayed reflex, muscle impedance, human body dynamics and external forces. The feedforward forces are determined by the planned trajectory and the delayed reflexes. The motor noise, inheriting in motion generation, is considered as an extension of the previous model in [9]. The impedance depends on the torque caused by muscle activation and the time delays of the reflexes. Finally, the feedforward and feedback control units are combined to generate the muscle force acting on the human arm to encounter external forces.

This model fully shows the human motor learning process and greatly influences the subsequent studies [10]-[16], such as in [12], Yang *et al.* proposed a sliding term to update the feedforward torque and the impedance factors (stiffness and damping) to replace the effect of muscle contraction. Li *et al.* created a controller that takes into account not only feedforward force and impedance, but also the adjustment of the reference trajectory during interactions with an unknown environment [13]. This model is also used for impedance control of robotic upper limb exoskeletons [15], [16]. However, most previous adaptation methods are based on iterative learning control and adaptive control , where the actuators' torques are updated at fixed rates (e.g., α and β in [7]) to minimize force and position tracking errors. The rate of force decrease in the simulations is somehow different from that observed in reality in [17]. Some recently learning methods that have been successfully used in human activities monitoring [18] and robot manipulation [19], [20], are helpful to recognize and minimize the difference. The other case is that the learning rate is different in different force fields. The experimental results in [21] showed that the absolute errors decrease more slowly in the divergent force (DF) field than in the velocity-dependent force (VF) field, which normally converge exponentially within 10 trails.

For the above properties in experiments, this paper proposes a new learning and control framework based on Broad Learning System (BLS) that the learning process is realized by increasing the number of feature nodes or enhancement nodes and linking between different layers to increase or decrease learning rate of the force and impedance, which is fully different from previous research. Referring to the models from [5] to [13], especially [8] and [12], we construct the motor learning and control diagram for torque and impedance in Fig. 2. Referring to [12], a sliding tracking error term is built for updating feedforward force. The difference of the proposed framework from other iterative learning methods is that the feedforward force is estimated using feature expressions and their networks, which is similar to feedforward model based on RBF neural network in [22]. As introduced in [24] and [8], the joint stiffness matrix increases with torque and muscle activation. Then, the neural feature terms in the feedforward torque block are transformed by adding the noise terms to describe contractile effect of the muscle. The impedance is expressed by the linear combination of enhancement nodes (the idea described by BLS) that is achieved by the transformations of feature nodes from the feedforward torque. The impedance changes are realized by increasing the number of enhancement nodes and updating the weights according to the tracking errors. The feedback torque is then achieved based on newly updated impedance, and the feedforward torque and feedback torque are combined to generate muscle forces (as in Fig.1(b)). Following [12], the inverse dynamics model is also used for robot control. In Fig. 2, the learning process is coloured with a grey background and calculated by BLS, but the rules for updating connections come from the biomimetic models.

The reminder of this paper is organized as follows: Section II introduces briefly BLS and the force and impedance learning framework based on BLS. The topics, such as

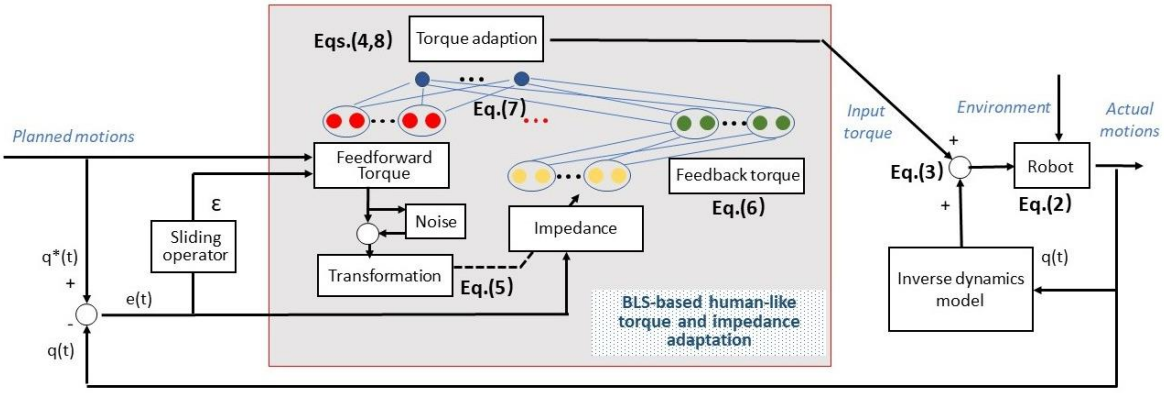

Fig. 2. Learning and control diagram of BLS-based human-like torque and impedance adaptation

impedance and feedback torque learning and nonlinear learning process etc., are discussed and compared with the previous models. Section III makes three groups of comparative simulations to compare the trajectory tacking effect, robustness for varying force fields and changing continuity of the force and trajectory in a complex environment. An experiment is taken to verify its effectiveness in actual interaction with the unstable environment. Section IV makes a conclusion and prospect for future work.

II. Broad Learning Based Force and Impedance Adaptation

A. Broad learning systems

BLS is proposed based on Random Vector Functional-Link Neural Network (RVFLNN) and has advantages of eliminating the disadvantage of long training process [23] etc.. BLS is a flat network where the original inputs are transferred to the feature layer as "mapped features" and the structure is extended in a wide sense in the "enhancement nodes" $H_i, i=1,...,n$ (shown in Fig. 3). In recent years, BLS has been used to control robots and micro-air vehicles [25] and system identification [26].

In Fig. 3, $\mathbf{X}$ is the input vector and $\mathbf{Y}\in\mathbf{R}^{N\times C}$ is the output variable of the network, where N is the number of feature mapping and C is the dimension of the network's output. To explore the hidden features of the input data, the feature mappings are expressed as $\mathbf{Z}_i=\phi_i\left(\mathbf{X}^T\mathbf{W}_{ei}+\boldsymbol{\beta}_{ei}\right),\ i=1,2,...,n$, where ϕ_i is a transformation function and $\mathbf{W}_{ei}$ and $\boldsymbol{\beta}_{ei}$ are sampled randomly from the distribution density.

Setting $\mathbf{Z}^n=\left[\mathbf{Z}_1,\mathbf{Z}_2,...,\mathbf{Z}_n\right]$, and the j*th* enhancement term H_j of the functional linking networks is generated by a linear transformation function ξ_j of $\mathbf{Z^n}$, similar to ϕ_i, set as $\mathbf{H}_j=\ \xi_j\left(\mathbf{Z^{n}}^T\mathbf{W}_{hj}+\boldsymbol{\beta}_{hj}\right), j=1,...,m$,where $\mathbf{W}_{hj}$ and $\boldsymbol{\beta}_{hj}$ are randomly generated sampling data from distribution density.

The linked network is expended by

$$\begin{aligned}\mathbf{Y}&=\left[\mathbf{Z}_1,...,\mathbf{Z}_n\middle|\xi_1\left(\mathbf{Z}^n\mathbf{W}_{h1}+\boldsymbol{\beta}_{h1}\right),...,\xi_m\left(\mathbf{Z}^n\mathbf{W}_{hm}+\boldsymbol{\beta}_{hm}\right)\right]^T\mathbf{W}\\&=\left[\mathbf{Z}_1,...,\mathbf{Z}_n\middle|\mathbf{H}_1,...,\mathbf{H}_m\right]^T\mathbf{W}\\&=\left[\mathbf{Z}^n\middle|\mathbf{H}^m\right]^T\mathbf{W}\\&=\left(\mathbf{A}^{n+m}\right)^T\mathbf{W}\end{aligned},\qquad(1)$$

where $\mathbf{X}^N=\left[\mathbf{X}\middle|x\right]$ represents the matrix $\mathbf{X}$ is extended in row by x to achieve a new matrix $\mathbf{X}^N$. The fitting accuracy of the network is improved by inserting additional feature nodes and enhancement nodes to approach to the desired output $\mathbf{Y}^*$ [23] .

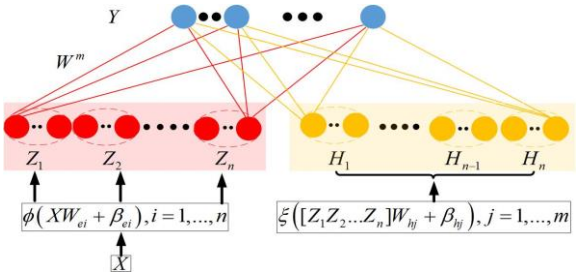

Fig.3 Illustration of BLS

B. BLS-based force and impedance adaptation framework

The robot system is described as

$$\mathbf{M}(q)\ddot{q}+\mathbf{C}(q,\dot{q})\dot{q}+\mathbf{G}=\boldsymbol{\tau}-\boldsymbol{\tau}_f+\boldsymbol{\tau}_v,\qquad(2)$$

where $q\in R^n$ represents and the simplification of joint $q(t)$ at time $t\in R^+$, $\mathbf{M}(q)\in R^{n\times n}$ is the inertia matrix, $\mathbf{C}(q,\dot{q})\in R^{n\times n}$ is the Coriolis and centrifugal torque matrix, and $\mathbf{G}\in R^n$ is the gravitational torque. $\boldsymbol{\tau}$ represents control torque and $\boldsymbol{\tau}_f$ is the external torque affected by the environmental force $\mathbf{F}_e$ that satisfies $\boldsymbol{\tau}_f=\mathbf{J}^T(q)\mathbf{F}_e$, where $\mathbf{J}^T(q)$ is a Jacobian matrix, and $\boldsymbol{\tau}_v$ is the noise term satisfying $\boldsymbol{\tau}_v\le\bar{\mathbf{v}}<\infty$ [12]. Set q^* as the reference trajectory to q, then the position error is $e=q^*-q$ and velocity error is $\dot{e}=\dot{q}^*-\dot{q}$ and the sliding term in Fig. 2 $\varepsilon\equiv e+\kappa\dot{e},\ \kappa>0$ is the tracking error commonly used in robot control [8] and [12].

The control torque $\boldsymbol{\tau}$ is designed as

$$\boldsymbol{\tau}=\underbrace{\mathbf{M}\ddot{q}^*+\mathbf{C}\dot{q}^*+\mathbf{G}-sign(\varepsilon)\bar{\mathbf{v}}}_{\text{Robot dynamics compensation}}+\underbrace{\boldsymbol{\tau}_a}_{\text{Torque adaptive}},\qquad(3)$$

where $\boldsymbol{\tau}_a$ is the output torque calculated by BLS, and $\ddot{q}^*$ and $\dot{q}^*$ are acceleration and velocity of the bounded periodic reference joint q^*, satisfying $q^*(t)=q^*(t-T)<\infty,\ T>0$, and *sign*(*) is the sign function that is defined component wise.

Torque τ is created in a similar expression as eq. (3) in [12] that consists of two parts: the former is for compensating robot dynamics based on $q^*(t)$ and bounded noise $\boldsymbol{\tau}_v$, and $\boldsymbol{\tau}_a$ is composed of the feedforward and feedback torques.

The feature nodes in the feedforward torque are calculated by the feature mapping function $\mathbf{s}_i=\phi_i\left(\mathbf{W}_{ei}\varepsilon+\boldsymbol{\beta}_{ei}\right)$, which is a linear function or a Gaussian function $\mathbf{s}_i=\exp\left[-\left(\varepsilon-\mu_i\right)^T\left(\varepsilon-\mu_i\right)/\eta_i^2\right], i=1,2,...,l$, where l is the number of function, and $\boldsymbol{\mu}_i=\left[\mu_{i1},\mu_{i2},...,\ \mu_{im}\right]\in R^m$ is a vector consisted of the center of each receptive field and η_i is the variance. Defining $\mathbf{S}(\varepsilon)=\left[\mathbf{s}_1,\mathbf{s}_2,...,\mathbf{s}_l\right]$, the ideal weight vector $\mathbf{W}^*$ is defined as the optimal value of $\mathbf{W}'$ that could minimize the approximation errors as

$$\mathbf{W}^{*} \overset{def}{=} \arg\min_{W' \in R^{l}} \left\{ \sup \left| \bar{\mathbf{F}} - \mathbf{S}(\varepsilon)^{T} \mathbf{W}' \right| \right\}, \tag{4}$$

where $\bar{\mathbf{F}}$ is desired feedback torque that can be described as the reflexes in [8] or the descending feedforward motor command from the CNS [7]. Kadiallah and Franklin *et al.* utilized radial-basis function neural network (RBFNN) learning for learning feedforward muscle activity in muscle coordinates by creating a cost function for combining the cost for movement feedback error and activation [17]. The usage of (4) is similar to the RBFNN-based feedforward model in [17], but the muscle visco-elasticity part is based on constant stiffness and constant damping factor and does not reveal the relationship between feedforward torque and impedance.

For this problem, [7] built a model based on the assumption that the intrinsic stiffness increases linearly with the motor command. In [12] and [13], the stiffness and damping matrices are adapted based on the function of ε, e and $\dot{e}$ with a varying forgetting factor of learning. In [24], the stiffness in the impedance model is determined by the muscle activation. Following [27], the feedback force is determined by each muscle stiffness and the number of activated muscles. Therefore, the impedance is calculated in two steps. In the first step, stiffness $\mathbf{K}_j(t)$ and damping factor $\mathbf{D}_j(t)$ are created for each enhancement node (like a muscle fiber):

$$\begin{cases} \mathbf{K}_j(t) = \varphi_j^{k}\left(\mathbf{S}(\varepsilon)\right) + \gamma\varphi_j^{k}\left(\Delta\mathbf{S}(\varepsilon)\right) + \boldsymbol{\sigma}_j^{k} \\ \mathbf{D}_j(t) = \varphi_j^{d}\left(\mathbf{S}(\varepsilon)\right) + \gamma\varphi_j^{d}\left(\Delta\mathbf{S}(\varepsilon)\right) + \boldsymbol{\sigma}_j^{d} \end{cases} \tag{5}$$

where $\varphi_j^{k}(*)$ is the transformation function for the jth enhance- ment node calculated by $\mathbf{S}(\varepsilon)$ from the feedforward torque. $\Delta\mathbf{S}(\varepsilon) = \mathbf{S}(\varepsilon)\big|_{t} - \mathbf{S}(\varepsilon)\big|_{t-T}$ represents changes of $\mathbf{S}(\varepsilon)$ during the learning process in (4), γ is a constant factor to ensure stability of the learning results and $\boldsymbol{\sigma}_j^{i}, i = k, d$ are noise terms. Each enhancement node generates feedback torque $\boldsymbol{\tau}_j^{f}$ that is expressed in an impedance form as

$$\begin{aligned} \boldsymbol{\tau}_j^{f} &= \mathbf{K}_j(t)e + \mathbf{D}_j(t)\dot{e} \\ &= \varphi_j\left(\mathbf{S}(\varepsilon), \Delta\mathbf{S}(\varepsilon), \boldsymbol{\sigma}_j^{k}, \boldsymbol{\sigma}_j^{d}, \gamma, e, \dot{e}\right) \end{aligned}, \tag{6}$$

where $\varphi_j(*)$ is a nonlinear transformation for the term $\mathbf{S}(\varepsilon)$ and its modification $\Delta\mathbf{S}(\varepsilon)$. We can get the enhancement group containing q feedback force terms as $\mathbf{F}^{q} = \left[\mathbf{F}_1, \mathbf{F}_2, ..., \mathbf{F}_q\right]$. The second step is modifying the number of enhancement node and weights of the exist feature and enhancement nodes. As the toque adaptation is the combined effect of feedforward torque and feedback torque. We can use (7), similar to (1), to express the torque $\boldsymbol{\tau}_a$ as

TABLE I
DEFINITIONS OF DIFFERENT FORCE FIELDS

Force field	Expressions
DF	$\mathbf{F}_{DF} = \begin{cases} \begin{bmatrix} 100 & 0 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix}, x \in (-0.2, 0.2) \\ 0 \qquad\qquad , otherwise \end{cases}$
CF	$\mathbf{F}_{CF} = \begin{bmatrix} 0 \\ 10 \end{bmatrix}$
VF	$\mathbf{F}_{VF} = K_v\dot{x}, K_v = -\begin{bmatrix} 13 & -18 \\ 18 & 13 \end{bmatrix}$
P-DF	$\mathbf{F}_{PDF} = 100\left(\lVert\Delta X\rVert - 0.12\right)\dfrac{\Delta X}{\lVert\Delta X\rVert}, \Delta X \equiv \begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} x \\ y \end{bmatrix}^{c}$
MF	$\mathbf{F}_{MF} = \begin{cases} \begin{bmatrix} 100 & 0 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, & x \in (0, 0.2) \\ & y \in (0.45, 0.6) \\ \begin{bmatrix} 0 \\ 10 \end{bmatrix}, & x \in (0, 0.2) \\ & y \in (0.2, 0.45] \\ -\begin{bmatrix} 13 & -18 \\ 18 & 13 \end{bmatrix}\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix}, & x \in (-0.2, 0] \\ & y \in (0.45, 0.6) \\ 100\left(\lVert\Delta X\rVert - 0.12\right)\dfrac{\Delta X}{\lVert\Delta X\rVert}, & x \in (-0.2, 0] \\ & y \in (0.2, 0.45] \\ X = [x, y]^{T}, \ \Delta X \equiv X - X^{c} \end{cases}$

$$\begin{aligned} \boldsymbol{\tau}_a &= \Big[\mathbf{s}_1(\varepsilon), ..., \mathbf{s}_l(\varepsilon) \Big| \varphi_1\left(\mathbf{S}(\varepsilon), \Delta\mathbf{S}(\varepsilon), \boldsymbol{\sigma}_1^{k}, \boldsymbol{\sigma}_1^{d}, \gamma, e, \dot{e}\right), ..., \\ &\quad \varphi_q\left(\mathbf{S}(\varepsilon), \Delta\mathbf{S}(\varepsilon), \boldsymbol{\sigma}_q^{k}, \boldsymbol{\sigma}_q^{d}, \gamma, e, \dot{e}\right)\Big]^{T}\left[\mathbf{W}' \middle| \mathbf{W}^{q}\right] \\ &= \left[\mathbf{S}(\varepsilon) \middle| \boldsymbol{\tau}_1^{f}, ..., \boldsymbol{\tau}_q^{f}\right]^{T} \mathbf{W}^{A} \\ &= \left(\mathbf{A}^{l+q}\right)^{T} \mathbf{W}^{A} \end{aligned}, \tag{7}$$

for minimizing the cost function:

$$\mathbf{W}^{A} \overset{def}{=} \arg\min_{W' \in R^{l}} \left\{ \sup \left| \bar{\boldsymbol{\tau}}_f - \mathbf{A}^{l+q}\mathbf{W}^{A'} \right| \right\}, \tag{8}$$

based on the calculation of $\mathbf{W}^{*}$ in (4), where $\bar{\boldsymbol{\tau}}_f$ represents the external torques calculated by the inverse robot dynamics at time $t - T, t \in [0, T)$, where T is the periodic time span.

C. Comments

1) **Impedance and feedback torque learning**
In [7], the feedforward force is described as the common effect of all the muscle fibers. In this paper, we mainly use connections of feature nodes and enhancement nodes to imitate muscle contraction mechanism in Fig. 1. First, the impedance is affected by the feedforward

torques and their noises, just as enhancement nodes are transferred from the feature nodes. Second, each enhancement node is considered as a muscle fiber and generates separate stiffness, damping, and contractile force in (5) and (6). Then, the product $\left[\boldsymbol{\tau}_1^f, \boldsymbol{\tau}_2^f, ..., \boldsymbol{\tau}_q^f\right]^T \mathbf{W}^q$ in (7) can be seen as the combined effect of all activated muscles. Lastly, but most importantly, the number of enhancement nodes increases with tracking demands. According to BLS, accuracy is improved by inserting additional feature nodes and enhancement nodes, which is similar to the contractile mechanism that impedance is increased by activating more muscle fibers.

2) **Learning process**

In the previous studies, e.g. [12], [13], the stiffness and damping are updated by a decayed factor and tracking errors. While in (5), $\mathbf{K}_j(t)$ and $\mathbf{D}_j(t)$ are calculated by using different transformation functions for each muscle. Then we can get $\boldsymbol{\tau}^f = \left[\boldsymbol{\tau}_1^f, \boldsymbol{\tau}_2^f, ..., \boldsymbol{\tau}_q^f\right]^T \mathbf{W}^q = \sum_{j=1}^{q} \left(\boldsymbol{\tau}_j^f\right)^T \mathbf{W}_j^q$, and the equivalent stiffness $\bar{\mathbf{K}}(t) = \sum_{j=1}^{q} \left(\mathbf{K}_j(t)\right)^T \mathbf{W}_j^q$ and the damping $\bar{\mathbf{D}}(t) = \sum_{j=1}^{q} \left(\mathbf{D}_j(t)\right)^T \mathbf{W}_j^q$. Due to φ_i^k and φ_j^k, φ_i^d and φ_j^d can be different functions for $i \neq j$, and the impedance factors satisfy $\mathbf{K}_i(t) \neq \mathbf{K}_j(t)$ and $\mathbf{D}_i(t) \neq \mathbf{D}_j(t)$ to imitate the difference in muscle fiber activations. The BLS-based learning structure takes advantages of sparse autoencoder characteristics to obtain better features and the stiffness and damping adaptation is realized by adding new enhancement nodes and updating weights by linear inverse functions in [23], which causes a nonlinear learning process. Moreover, referring to the control structure in [12], the proposed framework offers some advantages: no force sensor, control in joint and Cartesian space, and adaptive control with unknown parameters of the robot dynamics model .

III. Simulations and Experiment

There are three groups of comparative simulations. The first two simulations are based on the dataset provided in [27] and on https://www.imperial.ac.uk/human-robotics/software /, which are carried out on a planar arm using the two joint model of human arm/robot that is detailed in [8]. In [8], there are two kinds of force field: a position dependent divergent force (DF) field and a velocity dependent external force (VF) field. In [5], the authors defined two other force fields:

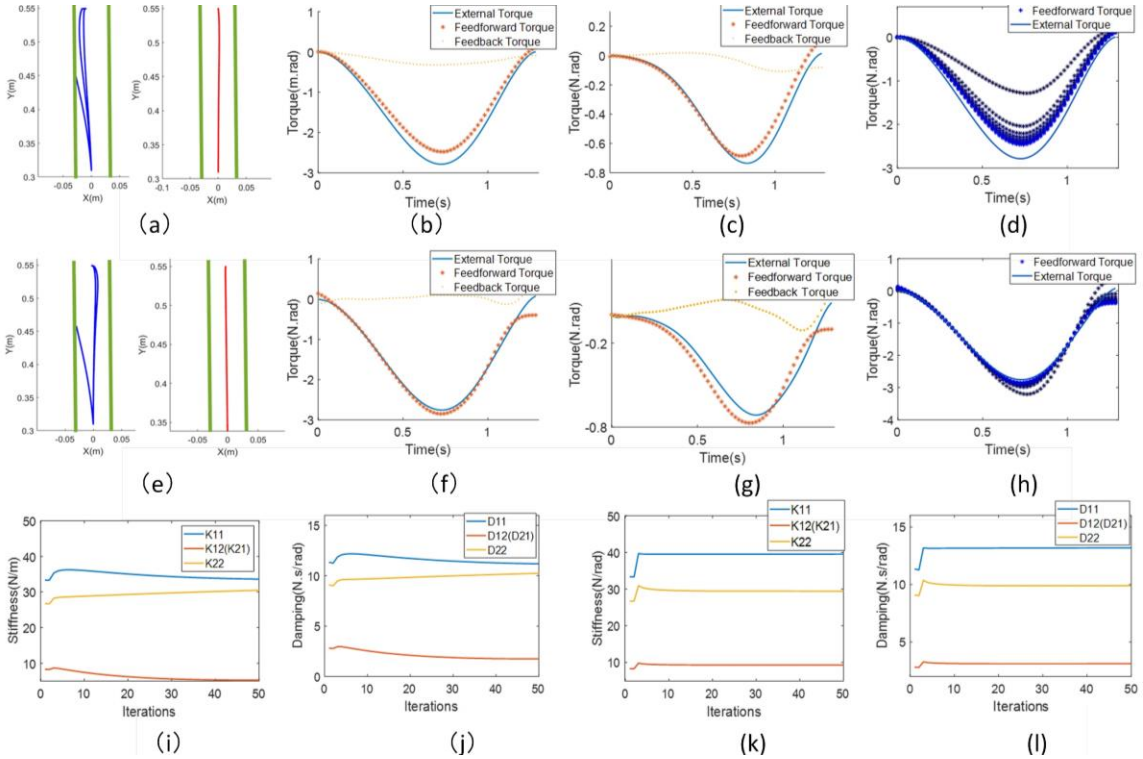

Fig. 4. Simulation of adaptation to unstable dynamics as in velocity dependent external force (VF) field. **(a)**. Trajectories in the first three iterations and last three iterations in [8]; **(b) and (c).** Feedback and feedforward joint torques and the torque affected by external forces in [8]; **(d)**. Evolution of feedforward torque in [8]; **(e).** Trajectories in the first three iterations and last three iterations in the proposed method; **(f) and (g).** Feedback and feedforward joint torques and the torque affected by external forces in the proposed method; **(h)**. Evolution of feedforward torque in the proposed method; **(i) and (j).** Stiffness and damping adaptation in [8]; **(k) and (i).** Stiffness and damping adaptation in the proposed method;

a constant interaction force (CF) field and a position-dependent divergent force (P-DF). In the third simulation, we will introduce a new mixed force (MF) field that combines the above four force fields to compare the effect of position tracking and force matching in a complex environment. The expressions of these force fields are listed in Table I.

A Line Tracking Task in the VF field

The first simulation takes place in the VF field and the comparison method is from [8]. The reference trajectory starts at $x_s = [0, 0.31]$ and ends at $x_d = [0, 0.55]$ with duration $T = 1.3s$. The parameters are $\kappa = 5$, $\gamma = 2$ and the feature mapping function is $\phi_i(x) = x$ and transformation functions for impedance factors are $\varphi_j^k(x) = 2xw_x$ and $\varphi_j^d(x) = 10xw_x$, where $w_x \in (0.5, 1)$ is a randomly selected factor. The iteration times are 50. Since the robust term is specific to robot control, it is not considered in the simulation as in [12]. Each iteration is completed within a periodic time and refines the parameters and calculations based on the results of the previous iteration.

The simulation results are shown in Fig.4. Fig.4 (a) shows the evolution of the trajectories in the first three and the last three trials. The final trajectory is almost a direct line between the start and the end as planned before adaptation. Figs.4 (b) and 4 (c) show that joint torques change with time and the feedforward torque eventually approaches that exerted by external forces. The adjustment process of the feedforward torque and impedance of the shoulder joint with the iterations is shown in Fig.4(d), (i) and (j). These variables can approach the final states rapidly but don't converge to the final state within 50 periods, which is not consistent with the experimental results that converge in exponential form within 10 trails [21].

Fig.4 (e) shows trajectories under the control of the proposed method. We can see that the trajectories converge within first three iterations and the final trajectories are

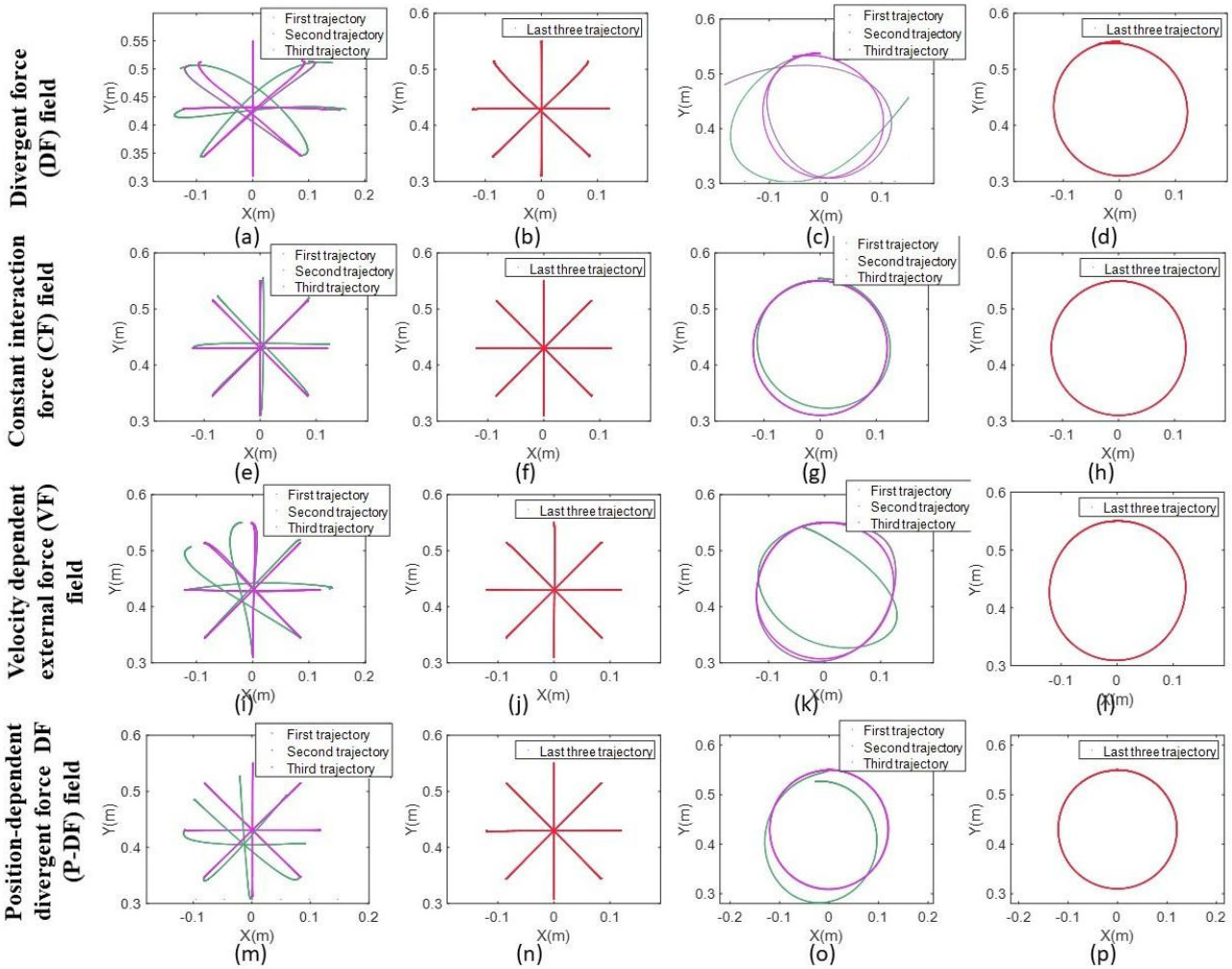

Fig. 5. Simulation of trajectory adaptation in different conditions in Table 1. **(a) to (d).** Trajectories in the first three iterations and last three iterations to follow line segments and (9) in DF field; **(e) to (h).** Trajectories in the first three iterations and last three iterations to follow line segments and (9) in CF field; **(i) to (l).** Trajectories in the first three iterations and last three iterations to follow line segments and (9) in VF field; **(m) to (p).** Trajectories in the first three iterations and last three iterations to follow line segments and (9) in P-DF field

straighter, which benefits from the faster and more efficient converging rate of the feedforward torque and impedance factors as shown in Fig.4 (h), Fig.4 (k), and Fig.4 (l) (less than 10 iterations). Similar to results in [8], the impedance factors increase initially and then decrease to the final value within the next trials.

B Tracking Tasks in Different Force Fields

The second simulation group is taken within four kinds of force fields in Table 1: DF, CF, VF and P-DF to complete the same task that the actuator move along four lines segments first:

(1) from $x_s=[-0.12,0.43]$ to $x_d=[0.12,0.43]$,

(2) from $x_s=[-0.085,0.345]$ to $x_d=[0.085,0.515]$,

(3) from $x_s=[0,0.31]$ to $x_d=[0,0.55]$,

(4) from $x_s=[0.085,0.345]$ to $x_d=[-0.085,0.515]$.

Then the actuator will track the circular trajectory

$$\begin{cases} x=-r\sin(kv(t)) \\ y=r\cos(kv(t))+0.43 \end{cases}, \tag{9}$$

with reference velocity profile as

$$v(t)=\frac{30t^2}{T^3}\left(1-2\left(\frac{t}{T}\right)+\left(\frac{t}{T}\right)^2\right), \tag{10}$$

where $r=0.12$, $k=15.2$ with period $T=4.5s$ and 50 iterations. The results are shown in Fig. 5. Each row shows trajectories generated in the same environment and each column compares those with different conditions. From Figs.5(b), (f), (j), (n) and (d), (h), (l), (p), it can be seen that the proposed controller responds well in different cases. After the first three trials, the trajectories are very close to the desired values and the final trajectories match the planned curves in every case. However, the final errors for DF and VF are slightly larger than those for CF and P-DF, which are mainly affected by the absolute magnitude of the external forces. The initial tracking effects are also different from each other. In the line tracking task, the initial position errors in the CF field are smaller than in the other three fields. The absolute errors in the DF field decrease more slowly than that in the VF and P-DF force fields, which are coincided with the experimental situations in [21]. The position errors to the trajectory ends in the P-DF force field are somewhat larger than in other fields. The circular tracking task achieves the similar conclusion. In the first trail, there is no any trajectory can complete a completely circle. In the third trail, the trajectories in the CF and P-DF force fields can realize end-to-end connections and are close to the circles. The worst performance is shown in Fig. 5(c) in the VF field, so more time is needed for the trajectory convergence and even after 50 periods, there are still some overlapping trajectory points.

C Circular Tracking Task in Complex Environment

In this simulation, we first set $\dot{x}=0.2m/s, \dot{y}=0.3m/s$ as an example and plot the force vectors of the MF field within the area $x\in(-0.2,0.2)$, $y\in(-0.3,0.6)$ in Fig.6 (a), which contains four force fields with four different background colours. The simulation uses the reference trajectory in (9),

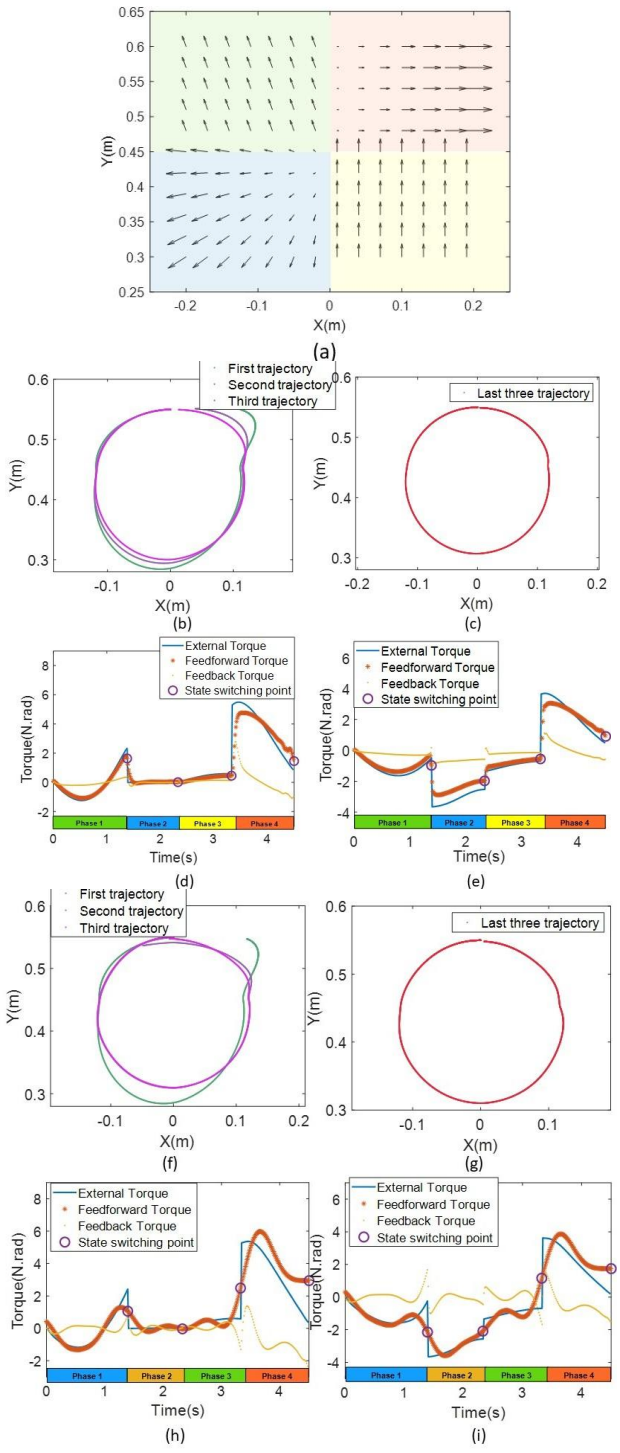

Fig. 6. Simulation of trajectory and torque adaptation in the MF field in Table 1 **(a)** Force field in four phases; **(b)&(c).** Trajectories in the first and last three periods in [8] **(d)& (e).** Joint torques and external torques in the final period in [8]; **(f)&(g).** Trajectories in the first and last three periods in the proposed method; **(h)&(i).** Joint torques and external torques in the final period in the proposed method

starting and ending both at $(0, 0.55)$. The adaptation is simulated in 50 iterations and $T = 4.5s$. Every trajectory passes sequentially through four force fields, numbered from Phase 1 to Phase 4, as indicated by the same colors in Fig.6 (d), (e), (h) and (i).

The simulation results of the controller in [8] for the circular tracking task are shown in Fig.6 (b) to Fig.6 (e). In Phase 2 and Phase 4, the position errors are the largest compared with the results in Phases 1 and Phases 3, which are consistent with the conclusion in Simulation 2. Fig 6.(d) and (e) show the torque changes in different phases. The feedforward torque has a large jump at the phase changing time that is affected by the changes of external torques, and some values are larger than 1 N. rad per step so that they will bring instability to system actuators. In the proposed method, the trajectories quickly converge to the desired values in all the phases except for the Phase 4, and the smoothness of the final trajectory is similar to that in Fig.6 (c). On the other hand, compared with Figs.6 (d) and (e), the continuity and uniformity of the feedforward force in Figs.6 (h) and (i) are much better, especially at the phase shifting moments between Phases 1 and 2, and Phases 3 and 4. This is consistent with the fact that the reactions of human muscles are smooth and continuous and shows that the proposed method is more suitable for processing tasks in a complex environment.

D Experiment

In this experiment, a robot arm is to follow a trajectory using a pen which is put in a forcing environment. As shown in Fig.7, the experiment is performed in an unstable environment with four elastic ropes attached to the corner rods of the wood board. A pen is fixed to the end of the Franka robot arm with a crank connecting the other ends of the elastic ropes to record the trajectory in the unstable environment. The Franka robot can record the contact force through the embedded force sensors as well as the joints and positions in the Cartesian space. Since the forces are provided by the elastic cables that are determined by the deformations along the pulling directions, the force field in the X-Y plane is shown in the zoomed figure in Fig.7. We can see that the pen and the robot arm, except for standing the center point, will be affected by an extra force to push the pen to the balanced center position. If we want the robot to draw a picture on the paper, the robot has to resist the extra force to complete the drawings.

Fig.8 shows the process and final drawings of demonstration are shown (a) through (d). The Human operator handled the pen to draw a square and the robot arm recorded the position while drawing. Then the robot tried to follow human demonstration by increasing the feedforward force and impedance factors. Here, we use the robot to record the trajectory in one trial and send the data to the simulation environment to generate the force and impedance for the next time to complete the trajectory tracking in the unstable force field. In Fig.8(e) to (f), the robot tried several times to finally follow the trajectory demonstrated by human operator.

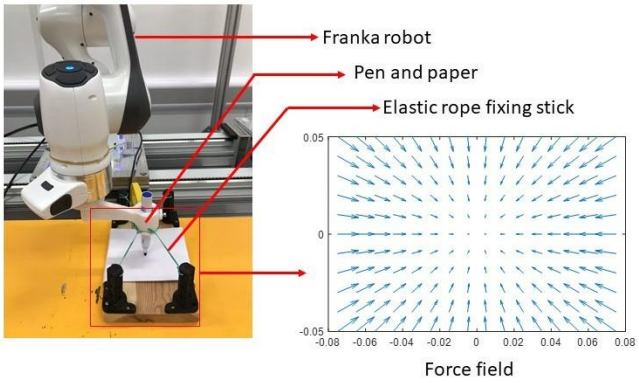

Fig. 7. Experimental setup with force field

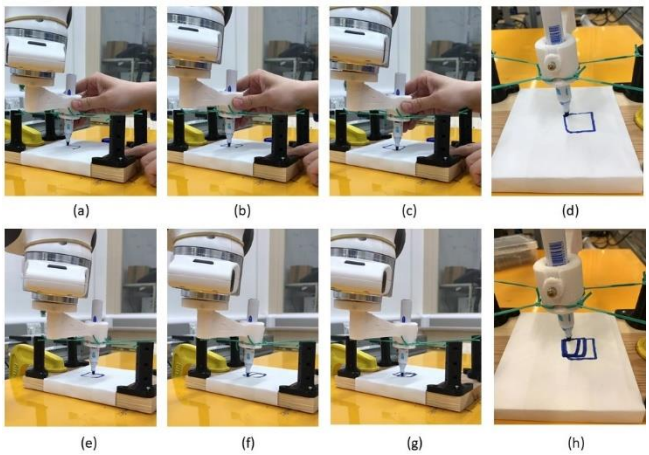

Fig. 8. Human demonstrations and robot force and impedance adaptation control **(a-c)** Demonstration through kinesthetic teaching **(d)** Teaching result **(e-g)** Robot force and impedance adaptation to track the demonstrated trajectory **(h)** trajectory adaptation results

E Discussion

The proposed learning and control framework transfers the iterative learning control into a fully neural network (NN) and the neural node linkage is endowed with biomimetic meanings, i.e., the impedances are affected by the feedforward torques by adding several enhancement nodes that have nonlinear relationships with the feature mapping functions. The above quantized simulation results show that the proposed framework has a faster convergence rate to the desired effect, a more robust response to the different environmental conditions, and better continuity and uniformity of the feedforward force in a complex force field. Nevertheless, there are some improvements for the current framework. For example, the learning rates of torques and impedances are not controllable, while in [8] they can be changed by choosing appropriate gain matrices. On the other hand, the difference between the force drop in the simulation and the force drop observed in reality is difficult to diminished through this framework. Nowadays, deep learning methods are possible to transfer motions and parameters from human demo-strations to robot manipulation [15], [16] to minimize the error between simulation and reality. In the future, it is desirable to combine the research of this work with

deep learning methods to develop a system for human skill learning and generalization.

IV. Conclusion

In this paper, a new biomimetic learning and control framework for force and impedance is proposed based on the theory of BLS. Following the central neural system and the human motor working mechanism, an incremental and explainable neural network with functional linkage is constructed for force and impedance adaptation and trajectory tracking in stable and unstable force fields. Some principles such as the impedance of muscle fibers and the amount of activated muscle fibers are reflected by the enhancement nodes in the flat neural network, which have hardly been modeled in previous research. Three groups of comparative simulations based on the open-access dataset are performed to verify the merits of tracking speed and accuracy, robustness to the differentiated force field environment, and smoothness and continuity of force changes in the complex unstable and stable environment. Our simulation results even prove to some extent the accuracy and effectiveness of the proposed method for some experimental phenomena in previous research. In future work, we will explore the optimization of the current framework and combine this research with human-like skill learning and generalization, and imitation learning.

Acknowledgment

This work was supported by the H2020 Marie Skłodowska-Curie Actions Individual Fellowship under Grant 101030691. Ning Wang is the corresponding author.

References

[1] C. Yang, J. Luo, Y. Pan, Z. Liu and C. -Y. Su, "Personalized Variable Gain Control With Tremor Attenuation for Robot Teleoperation," in IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 48, no. 10, pp. 1759-1770, Oct. 2018

[2] C. Yang, C. Zeng, Y. Cong, N. Wang and M. Wang, "A Learning Framework of Adaptive Manipulative Skills From Human to Robot," in IEEE Transactions on Industrial Informatics, vol. 15, no. 2,

[3] C. Yang, K. Huang, H. Cheng, Y. Li and C. -Y. Su, "Haptic Identification by ELM-Controlled Uncertain Manipulator," in IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 47, no. 8, pp. 2398-2409, Aug. 2017

[4] C. Yang, J. Luo, C. Liu, M. Li and S. Dai, "Haptics Electromyography Perception and Learning Enhanced Intelligence for Teleoperated Robot," in IEEE Transactions on Automation Science and Engineering

[5] D. W. Franklin, R. Osu, E. Burdet, M. Kawato, and T. E. Milner, "Adaptation to stable and unstable dynamics achieved by combined impedance control and inverse dynamics model," *Journal of neurophysiology*, vol. 90, no. 5, pp. 3270-3282, 2003.

[6] D. W. Franklin, E. Burdet, K. P. Tee, R. Osu, C.-M. Chew, T. E. Milner, and M. Kawato, "CNS learns stable, accurate, and efficient movements using a simple algorithm," *Journal of neuroscience*, vol. 28, no. 44, pp. 11165-11173, 2008.

[7] K. P. Tee, D. W. Franklin, M. Kawato, T. E. Milner, and E. Burdet, "Concurrent adaptation of force and impedance in the redundant muscle system," *Biological cybernetics*, vol. 102, no. 1, pp. 31-44, 2010.

[8] E. Burdet, K. P. Tee, I. Mareels, T. E. Milner, C.-M. Chew, D. W. Franklin, R. Osu, and M. Kawato, "Stability and motor adaptation in human arm movements," *Biological cybernetics,* vol. 94, no. 1, pp. 20-32, 2006.

[9] E., Burdet, K.P. Tee, C.M. Chew, J. Peters, and V.L. Bt, "Hybrid IDM/Impedance Learning in Human Movements." in Int Symposium on measurement, analysis and modeling of human functions, pp. 340-345, Sapporo, Japan ,2001.

[10] G. Ganesh, A. Albu-Schäffer, M. Haruno, M. Kawato, and E. Burdet, "Biomimetic motor behavior for simultaneous adaptation of force, impedance and trajectory in interaction tasks." in IEEE International Conference on Robotics and Automation, pp. 2705-2711,. 2010

[11] C. Zeng, C. Yang, and Z. Chen, "Bio-inspired robotic impedance adaptation for human-robot collaborative tasks," *Science China Information Sciences*, vol. 63, no. 7, pp.1-10. 2020.

[12] C. Yang, G. Ganesh, S. Haddadin, S. Parusel, A. Albu-Schaeffer, and E. Burdet, "Human-like adaptation of force and impedance in stable and unstable interactions," *IEEE transactions on robotics*, vol. 27, no. 5, pp. 918-930, 2011.

[13] Y. Li, G. Ganesh, N. Jarrassé, S. Haddadin, A. Albu-Schaeffer, and E. Burdet, "Force, impedance, and trajectory learning for contact tooling and haptic identification," *IEEE Transactions on Robotics*, vol. 34, no. 5, pp. 1170-1182, 2018.

[14] C. Yang, Y. Jiang, W. He, J. Na, Z. Li, and B. Xu, "Adaptive parameter estimation and control design for robot manipulators with finite-time convergence," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 10, pp.8112-8123, 2018.

[15] Z. Li, Z. Huang, W. He, and C.-Y. Su, "Adaptive impedance control for an upper limb robotic exoskeleton using biological signals," IEEE Transactions on Industrial Electronics, vol. 64, no. 2, pp. 1664-1674, 2016.

[16] Z. Li, C.-Y. Su, G. Li, and H. Su, "Fuzzy approximation-based adaptive backstepping control of an exoskeleton for human upper limbs," *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 3, pp. 555-566, 2014.

[17] R. A. Scheidt, D. J. Reinkensmeyer, M. A. Conditt, W. Z. Rymer, and F. A. Mussa-Ivaldi, "Persistence of motor adaptation during constrained, multi-joint, arm movements," *Journal of neurophysiology*, vol. 84, no. 2, pp. 853-862, 2000.

[18] W. Qi, H. Su and A. Aliverti, "A Smartphone-Based Adaptive Recognition and Real-Time Monitoring System for Human Activities," *IEEE Transactions on Human-Machine Systems*, vol. 50, no. 5, pp. 414-423, Oct. 2020.

[19] H. Su, W. Qi, Y. Hu, H. R. Karimi, G. Ferrigno and E. D. Momi, "An Incremental Learning Framework for Human-Like Redundancy Optimization of Anthropomorphic Manipulators," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 3, pp. 1864-1872, 2020.

[20] H. Su, Y. Hu, H.R. Karimi, A. Knoll, G. Ferrigno, and E. De Momi, "Improved recurrent neural network-based manipulator control with remote center of motion constraints: Experimental results," *Neural Networks*, vol. 131, pp.291-299. 2020.

[21] R. Osu, E. Burdet, D. W. Franklin, T. E. Milner, and M. Kawato, "Different mechanisms involved in adaptation to stable and unstable dynamics," *Journal of Neurophysiology*, no. 90, pp.3255-3269, 2003,

[22] A. Kadiallah, D. W. Franklin, and E. Burdet, "Generalization in adaptation to stable and unstable dynamics," *Plos one*, 2012, vol.7, no.10: e45075.

[23] C. P. Chen, and Z. Liu, "Broad learning system: An effective and efficient incremental learning system without the need for deep architecture," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 1, pp. 10-24, 2017.

[24] K. P. Tee, E. Burdet, C.-M. Chew, and T. E. Milner, "A model of force and impedance in human arm movements," Biological cybernetics, vol. 90, no. 5, pp. 368-375, 2004.

[25] H. Huang, T. Zhang, C. Yang, and C. P. Chen, "Motor learning and generalization using broad learning adaptive neural control," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 10, pp. 8608-8617, 2019.

[26] Z. Lu, N. Wang, and C. Yang, "A novel iterative identification based on the optimised topology for common state monitoring in wireless sensor networks*," International Journal of Systems Science*, vol. 53, no. 1, pp. 25-39,

[27] E. Burdet, D. W. Franklin, and T. E. Milner, Human robotics: neuro-mechanics and motor control: *MIT press*, 2013.