Proceedings of the ASME 2023
International Mechanical Engineering Congress and Exposition
IMECE2023
October 29-November 2, 2023, New Orleans, USA

IMECE2023-112415

DEEP NEURAL NETWORK BASED SATURATED ADAPTIVE CONTROL OF MUSCLES IN A LOWER-LIMB HYBRID EXOSKELETON

Jonathan A. Ting*, **Sujata Basyal, Brendon C. Allen**
Department of Mechanical Engineering
Auburn University, Auburn, Alabama 36830
Email: {jat0076,szb0233,bca0027}@auburn.edu

ABSTRACT

Hybrid exoskeletons are used to blend the rehabilitative efficacy and mitigate the shortcomings of functional electrical stimulation (FES) and exoskeleton-based rehabilitative solutions. This paper introduces a novel nonlinear controller that may potentially improve the rehabilitative efficiency of a lower limb hybrid exoskeleton by implementing four key features into the FES and exoskeleton controllers. First, the FES input to the user's muscles is saturated based on user preference to ensure user comfort. Second, rather than discarding the excess control effort from the saturated FES input, it is redirected into the exoskeleton's motor controller. Third, a safe deep neural network (DNN) is designed to estimate the unknown dynamics of the hybrid exoskeleton and the DNN is implemented in the FES controller to improve the control efficiency and tracking performance. Fourth, an adaptive update law is designed to estimate the unknown muscle control effectiveness to facilitate the implementation of the DNN. Lyapunov stability-based methods are used to generate real-time adaptive update laws that will train the adaptive estimate of the muscle effectiveness and the output layer weights of the DNN in real-time, ensure the effectiveness and safety of the controllers, and prove global asymptotic tracking of the desired trajectory.

1 Introduction[1]

Functional electrical stimulation (FES) is a rehabilitation technique that sends electrical pulses to a user's muscles, causing functional muscle contractions in limbs that are paralyzed or weakened due to a neurological condition or injury. FES has been used extensively in rehabilitation and has been proven to be highly effective [1–3]. Yet, despite the enormous rehabilitative potential of FES, the technique has one key flaw: FES evokes muscle fatigue at a higher rate than volitional muscle activation [4–6]. Because repetitive functional tasks are essential for neural plasticity and improving cardio-respiratory fitness, FES-induced muscle fatigue prevents the user from achieving the number of repetitions needed for clinically effective rehabilitation [7–9]. Therefore, to compensate for the faster onset of muscle fatigue induced by FES, hybrid exoskeletons were introduced to blend the rehabilitative efficacy and mitigate the shortcomings of FES and exoskeleton rehabilitative solutions. That is, the motors of an exoskeleton and stimulation from FES are used simultaneously to actuate the user's limb, allowing the user to achieve sufficient repetition for rehabilitative benefit, which may not be possible by using FES alone. In fact, hybrid exoskeletons have been documented to be effective during gait restoration and upper limb rehabilitation for users with spinal cord injuries [10–12]. Because FES may be uncomfortable for some users, hybrid exoskeletons are beneficial since they allow users to limit the FES input to the user's muscle whiles, still achieving proper tracking performance and enough repetitions for rehabilitative efficacy.

There have been previous attempts to saturate the FES input to the user's muscles [13–15]; however, these attempts include the FES saturation in an adhoc manner. Additionally, there have been some previous efforts that implement a saturated FES input into controller design like [16–18], but nearly all prior work simply discarded the excess FES effort. Therefore, similar to [16], this paper proposes a control architecture that sends the residual FES control effort to the motor controller when the FES input

*Address all correspondence to this author.

[1]This research is supported in part by NSF award number 2230971. Any opinions, findings and conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the sponsoring agency.

is saturated. Therefore, the motor controller will compensate for the saturated FES control effort, improving tracking performance and enhancing clinical rehabilitation effectiveness. Additionally, to slow muscle fatigue, the FES and motor controller will include adaptive DNN-based terms to approximate the uncertain exoskeleton dynamics.

Many Dynamic systems are highly complex, uncertain, hard to model, making it difficult to design an efficient controller that ensures tracking performance. Therefore, researchers have combined Lyapunov-based stability methods with neural network-based controllers to estimate the complex, unknown nonlinear dynamics [19–21]. Though neural networks have been proven to be quality function approximators, deep neural networks (DNNs) have been proven to be more capable of learning more complex dynamics than neural networks. [22, 23]. Although DNN's have better performance in approximating unknown dynamics DNN methods have traditionally lacked performance guarantees, hindering the adoption of DNN's in safety-critical systems. Furthermore, due to the computational power and slow training speed associated with training the multi-layer weights and biases of a DNN, the implementation of DNNs to a closed loop controller may not seem viable. Motivated to improve viability of DNN-based control, [24] and [25], have implemented a Lyapunov based real-time and offline DNN layer weight update approach to train the DNN to ensure consistent stability and tracking performance. To be specific, the outer-layer weights of the DNN are trained in real-time using Lyapunov based adaptive update laws; whereas, the inner layer weights are updated offline using batches of data collected during real-time experiments [26,27]. A current open problem is the implementation of a saturated DNN-based controller for a hybrid exoskeleton. The motivation of this paper it to integrate a DNN into a saturated hybrid exoskeleton controller to reduce fatigue and improve tracking performance and learning without sacrificing user comfort and safety.

Building off the controller design from [16] and [25], this paper develops a novel DNN-based saturated adaptive FES controllers and a robust motor controller for a lower-limb hybrid exoskeleton that will facilitate leg extension exercises. To be specific, to reduce the effect of muscular fatigue, the FES control input will be saturated and include adaptive DNN-based terms. If the FES input is saturated, the excess control effort is redirected into the motors of the exoskeleton, guaranteeing proper tracking performance and repetition. Furthermore, unlike [16], this work develops an adaptive estimate of the uncertain muscle control effectiveness to facilitate the adaptive FES control design. Additionally, a Lyapunov stability analysis is performed on the proposed control system to prove global asymptotic tracking.

2 Limb Model

The knee joint dynamics can be mathematically modeled as[2]

$$I\ddot{q} + P(q,\dot{q}) + G(q) + B\dot{q} + d(t) = \tau_H(t) + \tau_R(t), \quad (1)$$

where $q : \mathbb{R}_{\geq t_0} \to \mathbb{R}$ denotes the measurable, twice differentiable knee angle measured clockwise between the shank and the downward vertical axis, $t_0 \in \mathbb{R}_{\geq 0}$ is the initial time, $I \in \mathbb{R}_{>0}$ denotes the inertia of the lower limb about the knee joint, $P : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ denotes the passive viscoelastic torque associated with the user's joint stiffness and damping due to the muscle tendon complex, $G : \mathbb{R} \to \mathbb{R}$ denotes the gravitational force exerted on the lower limb, $B \in \mathbb{R}_{>0}$ denotes the viscous damping coefficient associated with exoskeletons' joint, $d : \mathbb{R}_{\geq t_0} \to \mathbb{R}$ denotes the unmodeled disturbances and dynamics, $\tau_H : \mathbb{R}_{\geq t_0} \to \mathbb{R}$ denotes the torque produced by the stimulated quadriceps femoris muscle (i.e., the human), and $\tau_R : \mathbb{R}_{\geq t_0} \to \mathbb{R}$ denotes the torque produced by the exoskeleton (i.e., the robot).Referencing (1), the torque produced by the stimulated quadriceps can be defined as

$$\tau_H(t) \triangleq B_h u_h(t), \quad (2)$$

where $u_h : \mathbb{R}_{\geq t_0} \to \mathbb{R}$ denotes the muscle control input, and $B_h \in \mathbb{R}$ denotes the unknown muscle control effectiveness term that maps the FES input to torque.[3]Conversely, the torque produced by the exoskeleton can be defined as

$$\tau_R(t) \triangleq B_r u_r(t), \quad (3)$$

where $u_r : \mathbb{R}_{\geq t_0} \to \mathbb{R}$ denotes the motor control input, and $B_r \in \mathbb{R}$ denotes the positive, known motor control effectiveness that maps motor input to torque.

Assumption 1: The product of the unknown muscle control effectiveness matrix and the control input can be linearly parameterized as

$$B_h u_h(t) = Y(t)\theta, \quad (4)$$

such that $Y : \mathbb{R}_{\geq t_0} \to \mathbb{R}$ is the known control input (i.e., $Y(t) \triangleq u_h(t)$), and $\theta \in \mathbb{R}$ is the unknown muscle control effectiveness constant (i.e., $\theta \triangleq B_h$).

Assumption 2: In the knee joint dynamics shown in (1), I, P, G, B, d have known bounds such that $c_i \leq I \leq c_I$, $|P(q,\dot{q})| \leq c_{P1} + c_{P2}|\dot{q}|$, $|G(q)| \leq c_G$, $B \leq c_B$, and $|d(t)| \leq c_d, \forall t$ where $c_i, c_I, c_{P1}, c_{P2}, c_G, c_B, c_d \in \mathbb{R}$ are known positive constants.

[2]For notational brevity, all explicit dependence on time, t, within the terms $q(t)$, $\dot{q}(t)$, $\ddot{q}(t)$ is suppressed.

[3]Due to changes in the muscle geometry, the muscle control effectiveness varies with motion of the knee. However, in this preliminary work, the control effectiveness is assumed to have a constant value. Future efforts will seek to modify the control development and stability analysis that enables the control effectiveness to vary with the knee angular position and velocity.

3 Control Development

3.1 Control Objective and Open Loop Error Dynamics

The control objective is to ensure that the knee joint angle tracks a sufficiently smooth desired position trajectory denoted by $q_d : \mathbb{R}_{\geq t_0} \to \mathbb{R}$, where q_d is a known, bounded, and twice differentiable signal such that $q_d, \dot{q}_d, \ddot{q}_d \in \mathcal{L}_\infty$. To ensure trajectory tracking, a tracking error represented by $e_1 : \mathbb{R}_{\geq t_0} \to \mathbb{R}$, that quantifies the deviation of the trajectory in (1) from the desired trajectory, is established. The tracking error is defined as[4]

$$e_1 \triangleq q_d - q. \tag{5}$$

To further facilitate the analysis, a filtered tracking error $e_2 : \mathbb{R}_{\geq t_0} \to \mathbb{R}$ is defined as:

$$e_2 \triangleq \dot{e}_1 + \alpha e_1, \tag{6}$$

where $\alpha \in \mathbb{R}_{>0}$ denotes a user-defined control gain. With the tracking error and the filtered tracking error defined, a composite error vector $z : \mathbb{R}^2 \to \mathbb{R}^2$ can be defined as:

$$z \triangleq \left[e_1,\, e_2 \right]^T. \tag{7}$$

The open-loop error dynamics can be obtained by taking the time derivative of (6), multiplying the time derivative of (6) with the inertia of the lower limb I, then adding and subtracting e_1 to the resulting product between the inertia of the lower limb and the derivative of the filtered tracking error, and substituting the dynamics in (1) and the product of the linearly parameterized control effectiveness and control input in (4) to yield

$$I\dot{e}_2 = f - Y\theta - B_r u_r + \Psi - e_1, \tag{8}$$

where $f : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is defined as

$$f \triangleq I\ddot{q}_d + P + G + B\dot{q}, \tag{9}$$

and $\Psi : \mathbb{R} \times \mathbb{R} \times \mathbb{R}_{\geq t_0} \to \mathbb{R}$ is defined as

$$\Psi \triangleq d + I\alpha\dot{e}_1 + e_1. \tag{10}$$

Since the constant muscle control effectiveness is unknown, an estimate of the muscle control effectiveness is necessary to facilitate controller development. The estimated control effectiveness can be obtained from the following equation:

$$\hat{B}_h(t)\, u_h(t) = Y(t)\, \hat{\theta}(t), \tag{11}$$

[4]For notational brevity, all functional dependencies are hereafter suppressed unless required for clarity of exposition.

where $\hat{B}_h, \hat{\theta}(: \mathbb{R}_{\geq t_0} \to \mathbb{R}_{>0}$ denote the estimate of the human control effectiveness constant since $\hat{\theta} \triangleq \hat{B}_h$. To facilitate analysis, the product of the estimated muscle control effectiveness and the control input terms are added and subtracted (i.e., $Y(t)\,\hat{\theta}(t) - \hat{B}_h u_h(t)$) into the open loop error dynamics in (8) to yield

$$I\dot{e}_2 = f - Y\tilde{\theta} - \hat{B}_h u_h(t) - B_r u_r + \Psi - e_1, \tag{12}$$

where the parameter estimation error $\tilde{\theta} : \mathbb{R}^2 \to \mathbb{R}$ is defined as

$$\tilde{\theta} \triangleq \theta - \hat{\theta}(t). \tag{13}$$

To facilitate the development of the DNN approximation, the desired exoskeleton dynamics $f_d : \Omega \to \mathbb{R}$ is defined as

$$f_d(x_d) \triangleq I\ddot{q}_d + P(q_d, \dot{q}_d) + G(q_d) + B\dot{q}_d, \tag{14}$$

where $x_d : \mathbb{R}_{\geq t_0} \to \Omega$ represents the composite desired trajectory that is defined as $x_d \triangleq [q_d, \dot{q}_d, \ddot{q}_d]^T$, and $\Omega \subset \mathbb{R}^3$ is a closed and bounded set. Now, f_d is added and subtracted into the open loop error system in (12) to yield

$$I\dot{e}_2 = S - Y\tilde{\theta} - \hat{B}_h u_h(t) - B_r u_r + \Psi - e_1 + f_d, \tag{15}$$

where $S = f - f_d$. Referencing (10) and (15), there exists constants $c_1, c_2 \in \mathbb{R}_{>0}$ that upper-bound Ψ and S such that $|\Psi + S| \leq c_1 + c_2||z||$ due to Assumption 2 and the Mean Value Theorem.

3.2 DNN Approximation and Update Policy

DNNs are capable of approximating continuous functions that lie on a compact set, which motivated the introduction of f_d in (15). Since $f_d : \Omega \to \mathbb{R}$, there exists an ideal, pre-trained DNN with ideal weights, biases, and activation functions such that the desired exoskeleton dynamics, f_d, can be represented using the universal approximation theorem for neural networks defined as

$$f_d(x_d) = W^T \sigma(\Phi(x_d)) + \varepsilon(x_d) \tag{16}$$

by using the universal approximation theorem for neural networks [28], where $W \in \mathbb{R}^{n \times 1}$ denotes the unknown, ideal output layer weights, $\sigma : \mathbb{R}^p \to \mathbb{R}^n$ denotes the unknown, ideal activation functions associated with the output layer of the DNN, $\Phi : \Omega \to \mathbb{R}^p$ denotes the unknown ideal DNN, $\varepsilon \in \Omega \to \mathbb{R}$ denotes the unknown DNN function approximation error, p denotes the number of final hidden layer neurons of the DNN, and n denotes the number of output layer neurons of the DNN. Referencing (16), the ideal DNN, Φ, can be defined as

$$\Phi \triangleq V_L \phi_L(V_{L-1} \phi_{L-1}(V_{L-2} \phi_{L-2}(\ldots V_1 \phi_1(x_d)), \tag{17}$$

where V_i and ϕ_i $\forall i \in [1, L]$

denote the hidden layer weights and activation function associated with the ideal DNN, respectively, and $L \in \mathbb{N}$ denotes the quantity of hidden layers associated with the ideal DNN.

Unlike most DNN-based control approaches, the DNN estimate in this work is trained using a multi-timescale approach. That is, the hidden layer weights and biases of the DNN are trained offline using traditional DNN training methods (i.e. gradient descent algorithm) with data sets collected prior to (i.e., pre-training) and concurrent to real-time execution, and the output layer weights are trained real-time using update laws that are developed using Lyapunov stability methods to ensure trajectory tracking performance and stability [25].Referencing (16), the DNN approximation of the desired hybrid exoskeleton dynamics, denoted by $\hat{f}_{d,i} : \Omega \to \mathbb{R}$, can be expressed as

$$\hat{f}_{d,i} = \hat{W}^T \hat{\sigma}_i \left(\hat{\Phi}_i(x_d) \right), \tag{18}$$

where $\hat{W} : \mathbb{R}_{\geq t_0} \to \mathbb{R}^{n\times 1}$ is the estimate of the ideal output layer weights, $\hat{\sigma}_i : \mathbb{R}^p \to \mathbb{R}^n$ is the i^{th} estimate of the ideal activation functions of the output DNN layer, and $\hat{\Phi}_i : \Omega \to \mathbb{R}^p$ is the i^{th} estimate of the ideal DNN, and $i \in \mathbb{N}$ represents the index for each DNN update. Note that pre-training the DNN provides $\widehat{\Phi}_1(\cdot)$ and $\widehat{W}(t_0)$. To develop a real-time update law for the output layer weights of the DNN, an error is established that quantifies the deviation between the ideal and estimate of the output layer weights. The error between the ideal and estimate of the output layer weights, denoted by $\tilde{W} : \mathbb{R}_{\geq t_0} \to \mathbb{R}^{n\times 1}$, is defined as:

$$\tilde{W}(t) \triangleq W - \hat{W}(t). \tag{19}$$

Assumption 3: Due to the universal function approximation theory of neural networks, there exists specific combinations of layer weights, biases, and activation functions that can approximate a function. Therefore, the unknown ideal output layer weights W, unknown ideal output layer activation functions $\sigma(\cdot)$, unknown ideal DNN $\Phi(\cdot)$, function approximation error $\varepsilon(\cdot)$, and the user defined activation functions $\hat{\sigma}(\cdot)$ can be upper-bounded with known constants $\overline{W}$, $\overline{\sigma}$, $\overline{\hat{\sigma}}$, $\overline{\varepsilon} \in \mathbb{R}_{>0}$ such that $\sup_{x_d(t)\in\Omega} \|W\| \leq \overline{W}$, $\sup_{x_d(t)\in\Omega} \|\sigma(\cdot)\| \leq \overline{\sigma}$, $\sup_{x_d(t)\in\Omega, \forall i\in\mathbb{N}} \|\hat{\sigma}_i(\cdot)\| \leq \overline{\hat{\sigma}}$,and $\sup_{x_d(t)\in\Omega} \|\varepsilon(\cdot)\| \leq \overline{\varepsilon}$.

3.3 Control Design and Closed Loop Error Dynamics

Based on the DNN approximation in (18) and the open loop error system in (15), the quadriceps femoral muscle controller is designed as

$$u_h \triangleq \text{sat}_\mu \left[\hat{B}_h^{-1} \hat{f}_{d,i} \right], \tag{20}$$

where sat_μ denotes a saturation function, with a limit of $\mu \in \mathbb{R}_{>0}$, is defined as

$$\text{sat}_\mu \left[\hat{B}_h^{-1} \hat{f}_{d,i} \right] \triangleq \begin{cases} \mu \text{sgn}\left(\hat{B}_h^{-1} \hat{f}_{d,i} \right) & \left| \hat{B}_h^{-1} \hat{f}_{d,i} \right| \geq \mu \\ \hat{B}_h^{-1} \hat{f}_{d,i} & \left| \hat{B}_h^{-1} \hat{f}_{d,i} \right| < \mu \end{cases}. \tag{21}$$

Recall that B_h is a positive constant. Consequently, $\hat{B}_h$ (i.e., $\hat{\theta}$) will be constrained to be a positive constant such that $\hat{B}_h^{-1}$ will always exist and be bounded.

Furthermore, the motor controller can be designed as

$$\begin{aligned} u_r \triangleq & B_r^{-1} \left(k_1 e_2 + (k_2 + k_3 \|z\|) \text{sgn}(e_2) \right) \\ & + B_r^{-1} \left(\hat{f}_{d,i} - \hat{B}_h u_h \right), \end{aligned} \tag{22}$$

where $k_1, k_2, k_3 \in \mathbb{R}_{>0}$ are positive, user defined control gains, and $\text{sgn}(\cdot)$ denotes the signum function. It is important to note that the difference between the estimated desired exoskeleton dynamics and the product of the estimated muscle control effectiveness term and the muscle control input in the motor controller (i.e., $\hat{f}_{d,i} - \hat{B}_h u_h$) will compensate for the residual control effort that results from the saturation term in (21). More precisely, the excess muscle control inputs are fed into the motor controller to ensure overall system performance. Based on the open loop error system in 15 and the subsequent stability analysis, the output layer weights of the DNN are updated using the an update law defined as

$$\dot{\hat{W}} \triangleq -\text{proj}\left(\Gamma \hat{\sigma}_i (\hat{\Phi}_i(x_d)) e_2 \right) \tag{23}$$

and the adaptive update law used for the estimation of the muscle control effectiveness term is defined as

$$\dot{\hat{\theta}} \triangleq -\text{proj}\left(\gamma Y e_2 \right), \tag{24}$$

where $\text{proj}(\cdot)$ denotes a smooth projection operator from [29] that ensures $\hat{\theta}$ and $\hat{W}$ are constrained within known bounds, and $\gamma \in \mathbb{R}_{>0}$ is a user defined learning rate that governs the pace at which the muscle control effectiveness estimate learns. Similarly, $\Gamma \in \mathbb{R}^{n\times n}$ is a user-defined diagonal matrix of learning rates that adjusts the learning rates of the estimates of the output layer weights of the DNN. After substituting the muscle controller in (20), the motor controller in (22), the ideal DNN approximation in (16), and the DNN estimate in (18) into the open loop error dynamics in (15), the closed loop error dynamics can be obtained as

$$\begin{aligned} I\dot{e}_2 = & -Y\tilde{\theta} - k_1 e_2 - k_2 \text{sgn}(e_2) - k_3 \|z\| \text{sgn}(e_2) \\ & + W^T \sigma(\Phi(x_d)) - \hat{W}^T \hat{\sigma}_i \left(\hat{\Phi}_i(x_d) \right) \\ & S + \Psi + \varepsilon(x_d) - e_1. \end{aligned} \tag{25}$$

It is important to note that the DNN performance will be improved if the DNN is pre-trained prior to controller operation. That is, all layer weights and biases of the DNN are initialized with optimal weights and biases based on dynamic data collected in previous simulations or experiments. To further improve the DNN performance, data is collected during controller operation to perform iterative offline updates (concurrent to real-time execution) of the inner layer DNN weights, and the output layer weights of the DNN are updated in real-time using the update law in (23) to ensure optimal tracking and stability. However, it should be noted that system performance is guaranteed even if the DNN is initially randomized although overall performance will be worse than cases when the DNN is properly pre-trained [25].

4 Stability Analysis

The stability of the DNN-based saturated adaptive control system is stated in the following theorem:

Theorem 1: Given the closed loop error dynamics in (25) that satisfies Assumptions 1-3, the control inputs in (20) and (22), and the adaptation laws in (23) and (24) ensure that the composite error vector defined in (7) (and hence the tracking errors defined in (5) and (6)) yields a global asymptotic tracking result in a sense that $\lim_{t\to\infty} e_1(t) = 0$ and $\lim_{t\to\infty} e_2(t) = 0$ provided that the following gain conditions are satisfied:

$$k_2 > c_1 + \bar{\varepsilon} + \overline{W}\left(\bar{\sigma} + \bar{\hat{\sigma}}\right), \qquad k_3 > c_2. \tag{26}$$

Proof: Let $V_L : \mathbb{R}^{3+n} \to \mathbb{R}$ be a candidate Lyapunov function defined as

$$V_L(\eta) = \frac{1}{2}e_1^2 + \frac{1}{2}Ie_2^2 + \frac{1}{2}\tilde{W}^T\Gamma^{-1}\tilde{W} + \frac{1}{2\gamma}\tilde{\theta}^2, \tag{27}$$

where $\eta : \mathbb{R}_{\geq t_0} \to \mathbb{R}^{3+n}$ is defined as $\eta = \left[\, e_1,\, e_2,\, \tilde{W}^T,\, \tilde{\theta}\,\right]^T$. For $t \in [t_0, \infty)$, let $\eta(t)$ be a Filippov solution to the differential inclusion $\dot{\eta} \in K[h](\eta)$, where $h(\eta) = \left[\, \dot{e}_1,\, \dot{e}_2,\, \dot{\tilde{W}}^T,\, \dot{\tilde{\theta}}\,\right]^T$ and $K[\cdot]$, defined in [30], is Filippov's differential inclusion operator. Due to the discontinuous sliding mode terms included in the motor controller in (22) and the DNN estimate in (18) being updated iteratively, time derivative of (27) exists almost everywhere (a.e) within $t_0 \in [t_0, \infty)$ such that $\dot{V}_L(\eta) \overset{\text{a.e.}}{\in} \dot{\tilde{V}}_L(\eta)$, where $\dot{\tilde{V}}_L(\eta)$ denotes the generalized time derivative of V_L along the Filippov trajectories of $\dot{\eta} = h(\zeta)$. Referencing (27), the generalized time derivative of V_L can be denoted as $\dot{\tilde{V}} \subseteq \bigcap_{\xi \in \partial V_L(\eta)} \xi^T \left[K[h]^T(\eta), 1\right]^T$ (according to [31, Equation 13]), where $\partial V_L(\eta)$ denotes the Clark's Generalized gradient of $V_L(\eta)$, and, since $V_L(\eta)$ is continuously differential in terms of η, Clark's generalized gradient of $V_L(\eta)$ can be redefined as $\partial V_L(\eta) = \nabla V_L(\eta)$, where ∇V_L denotes the standard gradient of V_L with respect to η and t.

After taking the generalized time derivative of the Lyapunov equation in (27), the closed loop error dynamics in (25), the filtered trajectory tracking error in (6), the parameter estimation error in (13), the error between the ideal output layer weights and estimated output layer weights in (19), the output layer adaptation law in (23), and the parameter estimation law in (24) are substituted into the generalized derivative of the Lyapunov function yielding

$$\begin{aligned}\dot{\tilde{V}}_L \subseteq\ & -\alpha e_1^2 + e_2 S - k_1 e_2^2 - k_2 e_2 K\left[\text{sgn}(e_2)\right] \\ & -k_3 \|z\|\, e_2 K\left[\text{sgn}(e_2)\right] + e_2 W^T \sigma(\Phi(x_d)) \\ & -e_2 \hat{W}^T K\left[\hat{\sigma}_i\left(\hat{\Phi}_i(x_d)\right)\right] + e_2 \varepsilon(x_d) \\ & +e_2 \Psi + \tilde{W}^T K\left[\hat{\sigma}_i(\hat{\Phi}_i(x_d))\right] e_2.\end{aligned} \tag{28}$$

Using (19) and adding and subtracting $e_2 W^T K\left[\hat{\sigma}_i(\hat{\Phi}_i(x_d))\right]$ into (28) yields

$$\begin{aligned}\dot{\tilde{V}}_L \subseteq\ & -\alpha e_1^2 - k_1 e_2^2 - k_2 e_2 K\left[\text{sgn}(e_2)\right] \\ & -k_3 \|z\|\, e_2 K\left[\text{sgn}(e_2)\right] + e_2\left(S + \Psi + \varepsilon(x_d)\right) \\ & +e_2\left(W^T \sigma(\Phi(x_d)) - W^T K\left[\hat{\sigma}_i(\hat{\Phi}_i(x_d))\right]\right).\end{aligned} \tag{29}$$

Using the fact that $|\Psi + S| \leq c_1 + c_2||z||$, Assumption 3, and the fact that $e_2 K\left[\text{sgn}(e_2)\right] = |e_2|$, (29) can be upper-bounded as

$$\begin{aligned}\dot{V}_L \overset{\text{a.e.}}{\leq}\ & -\alpha e_1^2 - k_1 e_2^2 - (k_3 - c_2)\|z\|\,|e_2| \\ & -\left(k_2 - c_1 - \bar{\varepsilon} - \overline{W}\left(\bar{\sigma} + \bar{\hat{\sigma}}\right)\right)|e_2|.\end{aligned} \tag{30}$$

By satisfying the conditions stated in (26), (30) can be further upper-bounded as

$$\dot{V}_L \overset{\text{a.e.}}{\leq} -\alpha e_1^2 - k_1 e_2^2 \leq -\min(\alpha, k_1)\|z\|^2. \tag{31}$$

Knowing that the Lyapunov function in (27) is positive definite and radially unbounded and that the time derivative of the Lyapunov function is negative semi-definite implies that $V_L(\eta) \in \mathcal{L}_\infty$, which implies that $\eta \in \mathcal{L}_\infty$. Due to the definition of η and the fact that $\eta \in \mathcal{L}_\infty$, it is implied that $e_1, e_2, \tilde{W}, \tilde{\theta} \in \mathcal{L}_\infty$. Because $e_1, e_2, \tilde{W}, \tilde{\theta} \in \mathcal{L}_\infty$, the relationships in (5), (13), and (19) implies that $q(t), \hat{\theta}(t), \hat{W}(t) \in \mathcal{L}_\infty$. If $q(t), \hat{\theta}(t), \hat{W}(t) \in \mathcal{L}_\infty$ then the relationship in (11) implies that $\hat{B} \in \mathcal{L}_\infty$. The definition of the muscle controller in (20) implies that $Y, u_h \in \mathcal{L}_\infty$. Assumption 2, the fact that $\hat{W}(t) \in \mathcal{L}_\infty$, and the definition in (18) implies that $\hat{\sigma}_i, \hat{f}_{d,i} \in \mathcal{L}_\infty$. Using the prior results and (22), it can be shown that $u_r \in \mathcal{L}_\infty$. Furthermore, by the extension of the LaSalle-Yoshizawa Theorem for non-smooth systems in [32] and [33] and because $V_L(\eta)$ is positive definite and $\dot{V}_L(\eta)$ is negative semi-definite, it can be shown that $-\min(\alpha, k_1)\|z(t)\|^2 \to 0$ as $t \to \infty$, which would imply that $e_1(t), e_2(t) \to 0$ as $t \to \infty$.

5 Conclusion

FES and motor controllers were developed for a leg extension hybrid exoskeleton. The FES controller included saturated feedforward DNN terms in an effort to slow the onset of muscle fatigue. A special feature of this work is that whenever the FES input is saturated, the excess is sent into the motors of the exoskeleton to improve tracking performance and repetition. To be specific, the motor controller uses robust feedback terms when actuating the user's limb under non-saturated conditions; but, when the FES is saturated, the motor include the excess DNN-based terms from the FES controller. To train the DNN, the output layer weights are updated in real-time using Lyapunov based adaptive update laws; whereas, the hidden layer weights are updated offline concurrent to real-time execution using traditional DNN training methods with data collected in real-time. Additionally, a nonsmooth Lyapunov-based stability analysis proves that the controllers yield a global asymptotic tracking result.

References

[1] Rebecca Martin, Cristina Sadowsky, Kimberly Obst, Brooke Meyer, and John McDonald. Functional electrical stimulation in spinal cord injury: From theory to practice. *Topics in spinal cord injury rehabilitation*, 18:28–33, 01 2012.

[2] Cesar Marquez Chin and Milos Popovic. Functional electrical stimulation therapy for restoration of motor function after spinal cord injury and stroke: a review. *BioMedical Engineering OnLine*, 19, 05 2020.

[3] O Sujith. Functional electrical stimulation in neurological disorders. *European journal of neurology : the official journal of the European Federation of Neurological Societies*, 15:437–44, 06 2008.

[4] Morufu Olusola Ibitoye, Nur Azah Hamzaid, Nazirah Hasnan, Ahmad Khairi Abdul Wahab, and Glen M. Davis. Strategies for rapid muscle fatigue reduction during fes exercise in individuals with spinal cord injury: A systematic review. *Plos One*, 2016.

[5] Maria Vromans and Pouran Faghri. Functional electrical stimulation-induced muscular fatigue: Effect of fiber composition and stimulation frequency on rate of fatigue development. *Journal of electromyography and kinesiology : official journal of the International Society of Electrophysiological Kinesiology*, 38:67–72, 11 2017.

[6] Martin Schmoll, Ronan Le Guillou, David Borges, Charles Fattal, Emerson Fachin-Martins, and Christine Azevedo-Coste. Standardizing fatigue-resistance testing during electrical stimulation of paralysed human quadriceps muscles, a practical approach. *Journal of NeuroEngineering and Rehabilitation*, 18, 01 2021.

[7] Hideaki Onishi. Cortical excitability following passive movement. *Physical Therapy Research*, 21, 11 2018.

[8] F. Quandt and F. C. Hummel. The influence of functional electrical stimulation on hand motor recovery in stroke patients: a review. *Exp. Transl. Stroke Med.*, 2014.

[9] Joanna Cholewa, Agnieszka Gorzkowska, Michal Szepelawy, Agnieszka Nawrocka, and Jaroslaw Cholewa. Influence of functional movement rehabilitation on quality of life in people with parkinson's disease. *Journal of physical therapy science*, 26:1329–1331, 09 2014.

[10] Antonio del Ama, Aikaterini Koutsou, Juan Moreno, Ana de los Reyes-Guzmán, Angel Gil-Agudo, and José Pons. Review of hybrid exoskeletons to restore gait following spinal cord injury. *Journal of rehabilitation research and development*, 49:497–514, 06 2012.

[11] C. A. Cousin. *Hybrid Exoskeletons For Rehabilitation: A Nonlinear Control Approach*. PhD thesis, University of Florida, 2019.

[12] Yi Long, Zhi-jiang Du, Weidong Wang, and Wei Dong. Development of a wearable exoskeleton rehabilitation system based on hybrid control mode. *International Journal of Advanced Robotic Systems*, 13, 10 2016.

[13] K. Kurosawa, R. Futami, T. Watanabe, and N. Hoshimiya. Joint angle control by FES using a feedback error learning controller. *IEEE Trans. Neural Syst. Rehabil. Eng.*, 13(3):359–371, September 2005.

[14] H. Kawai, M. J. Bellman, R. J. Downey, and W. E. Dixon. Tracking control for FES-cycling based on force direction efficiency with antagonistic bi-articular muscles. In *Proc. Am. Control Conf.*, pages 5484–5489, 2014.

[15] H. Kawai, M. Bellman, R. Downey, and W. E. Dixon. Closed-loop position and cadence tracking control for FES-cycling exploiting pedal force direction with antagonistic bi-articular muscles. *IEEE Trans. Control Syst. Tech.*, 27(2):730–742, February 2019.

[16] Jace B. Aldrich and Christian A. Cousin. Saturated adaptive control of antagonistic muscles on an upper-limb hybrid exoskeleton. In *2022 American Control Conference (ACC)*, pages 4397–4402, 2022.

[17] B. C. Allen, K. J. Stubbs, and W. E. Dixon. Saturated control of a switched FES-cycle with an unknown time-varying input delay. In *IFAC Conf. Cyber-Phys. Hum.-Syst.*, 2020.

[18] B. C. Allen, K. J. Stubbs, and W. E. Dixon. Position and cadence tracking of a motorized FES-cycle with an unknown time-varying input delay using saturated FES control. *Automatica*, 139, 2022.

[19] F. L. Lewis. Nonlinear network structures for feedback control. *Asian J. Control*, 1(4):205–228, 1999.

[20] P. M. Patre, W. MacKunis, K. Kaiser, and W. E. Dixon. Asymptotic tracking for uncertain dynamic systems via a multilayer neural network feedforward and RISE feedback control structure. *IEEE Trans. Autom. Control*, 53(9):2180–2185, 2008.

[21] P. Patre, S. Bhasin, Z. D. Wilcox, and W. E. Dixon. Composite adaptation for neural network-based controllers. *IEEE Trans. Autom. Control*, 55(4):944–950, 2010.

[22] Thomas Parisini and Riccardo Zoppoli. A receding-horizon

regulator for nonlinear systems and a neural approximation. *Automatica*, 31(10):1443–1451, 1995.

[23] Benjamin Karg and Sergio Lucia. Efficient representation and approximation of model predictive control laws via deep learning. *IEEE Transactions on Cybernetics*, 50(9):3866–3878, 2020.

[24] Max L. Greene, Zachary I. Bell, Scott Nivison, and Warren E. Dixon. Deep neural network-based approximate optimal tracking for unknown nonlinear systems. *IEEE Transactions on Automatic Control*, pages 1–8, 2023.

[25] R. Sun, M. Greene, D. Le, Z. Bell, G. Chowdhary, and W. E. Dixon. Lyapunov-based real-time and iterative adjustment of deep neural networks. *IEEE Control Syst. Lett.*, 6:193–198, 2022.

[26] G. Chowdhary and E. Johnson. A singular value maximizing data recording algorithm for concurrent learning. In *Proc. Am. Control Conf.*, pages 3547–3552, 2011.

[27] Girish Chowdhary, Tansel Yucelen, Maximillian Mühlegg, and Eric N. Johnson. Concurrent learning adaptive control of linear systems with exponentially convergent bounds. *Int. J. Adapt. Control Signal Process.*, 27(4):280–301, 2013.

[28] F. L. Lewis, S. Jagannathan, and A. Yesildirak. *Neural network control of robot manipulators and nonlinear systems*. CRC Press, Philadelphia, PA, 1998.

[29] Z. Cai, M. S. de Queiroz, and D. M. Dawson. A sufficiently smooth projection operator. *IEEE Trans. Autom. Control*, 51(1):135–139, January 2006.

[30] Brad E. Paden and Shankar S. Sastry. A calculus for computing Filippov's differential inclusion with application to the variable structure control of robot manipulators. *IEEE Trans. Circuits Syst.*, 34(1):73–82, January 1987.

[31] D. Shevitz and B. Paden. Lyapunov stability theory of nonsmooth systems. *IEEE Trans. Autom. Control*, 39 no. 9:1910–1914, 1994.

[32] N. Fischer, R. Kamalapurkar, and W. E. Dixon. LaSalle-Yoshizawa corollaries for nonsmooth systems. *IEEE Trans. Autom. Control*, 58(9):2333–2338, Sep. 2013.

[33] R. Kamalapurkar, J. A. Rosenfeld, A. Parikh, A. R. Teel, and W. E. Dixon. Invariance-like results for nonautonomous switched systems. *IEEE Trans. Autom. Control*, 64(2):614–627, February 2019.