Equilibrium Torque Control-Based Lower-Limb Exoskeleton Assistance With Memory-Enhanced Gait Prediction and Real-Time Learning

Wenlong Li, Yiming Fei, Hao Su, *Senior Member, IEEE*, Qi Li, and Yanan Li, *Senior Member, IEEE*

***Abstract*—The control of lower-limb exoskeletons plays a crucial role in determining the effectiveness of walking assistance, but how to generate a reference signal still poses a significant challenge. Many existing approaches involve offline training and classifiers or depend on prefabricated models, lacking the adaptability needed to support diverse users and real-time scenarios with varying gait cycles. Meanwhile, balancing intervention on human limbs between compliance and assistance during learning is still an open problem. To address these issues, this article proposes a real-time learning method for walking assistance without classifiers, automatically adapting to alterations in motion patterns. The control law, based on adaptive admittance control and the equilibrium state, ensures stable assistance during learning with intuitive parameter tuning and allows for switching of gait during assistance. Utilizing selective memory recursive least squares in a neural network enables rapid learning and precise prediction of human users' motion intention, without pretraining. Experimental results demonstrate that our approach achieves a prediction error within 6° after half a minute of learning with a prediction ahead time of 120 ms, outperforming classic approaches. The assistance performance is consistent despite varied control parameters, indicating a certain level of robustness.**

***Index Terms*—Equilibrium-based torque control (ETC), gait prediction, lower-limb assistance, real-time learning, selective memory recursive least square (SMRLS).**

Received 23 September 2024; revised 21 March 2025; accepted 3 June 2025. Date of publication 8 July 2025; date of current version 18 September 2025. This work was supported by Xiamen VNCOL Science and Technology Company, Ltd. This article was recommended by Associate Editor S. S. Ge. *(Corresponding author: Yanan Li.)*

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by Sciences and Technology Research Ethics Committee at the University of Sussex under Application No. ER/WL338/1.

Wenlong Li and Yanan Li are with the Department of Engineering and Design, University of Sussex, BN1 9RH Brighton, U.K. (e-mail: wl338@sussex.ac.uk; yl557@sussex.ac.uk).

Yiming Fei is with the College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China (e-mail: yimingfei@zju.edu.cn).

Hao Su is with the Laboratory of Biomechatronics and Intelligent Robotics, Tandon School of Engineering, New York University, New York, NY 11201 USA (e-mail: hao.su@nyu.edu).

Qi Li is with the VNCOL Science and Technology Company Ltd., Xiamen 361005, China (e-mail: liqi@vncol.com).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TSMC.2025.3580690.

Digital Object Identifier 10.1109/TSMC.2025.3580690

I. Introduction

Lower-limb exoskeletons (LLEs) are instrumental in augmenting human gait, providing crucial support for individuals with diverse motor impairments [1]. Partial exoskeletons are typically designed for applications involving intricate tasks, such as augmenting swift movements or handling vigorous activities [2]. Therefore, it is essential to anticipate and accurately estimate the user's intentions in advance. Meanwhile, the design of control laws significantly influences the device's ability to interact stably and purposefully with both the user and the environment [1], [3], [4].

According to the function of controllers, they are usually classified into three levels: 1) high; 2) mid; and 3) low [3]. The high-level controller is responsible for recognizing the user's intentions and subsequently translating these intentions into a reference by the mid-level controller, such as desired position and velocity. This reference then serves as the input of the low-level controller. Movement recognition is one of the most popular methods for the high-level controller which automatically adjusts its behavior based on the user's movements or intended movements [5], [6], [7], [8]. To generate a motor command, which is defined as the detection sublayer in the mid-level controller [1], current approaches predominantly include trajectory tracking, assist-as-needed [4] and model-based methods. Although trajectory tracking is still the most commonly employed strategy among them [4], other strategies can be combined with it [9], [10], [11]. Generation of a reference can be achieved either offline or online. Offline methods typically rely on extensive pretraining or theoretical models, demanding significant upfront effort and other methods to achieve adaptability [10], [12]. Sim2real with reinforcement learning is also a popular approach in offline training [13], which relies on learning-in-simulation with musculoskeletal and exoskeleton models. However, offline training methods may be difficult to respond to unexpected variations and highly dependent on the generalization and adaptability of models used in simulation. In comparison, online methods, such as adaptive-frequency oscillator or iterative learning control (ILC) require that the trajectory be periodic or repetitive [14], [15], which limits their generalization capability. Neural network-based learning and estimation is also a popular method for high-level controller, but its training is usually computationally demanding. For example, by comparing various neural network approaches, the computation time for each sample ranged from approximately 37.6 to 133.5 ms in [16]. In practice, the computation frequency, communication frequency, and control frequency limit each other and lower control frequency leads to unnatural gait or oscillations during real-time assistance. Consequently, a major challenge lies in developing methods to simultaneously train the model and provide assistance in real-time, achieving both adaptability and efficiency during operation.

In neural network-based gait prediction methods, frequent transitions among different gait behaviors are required by the high-level controller during locomotion [3], [17], [18], [19], [20], which are usually achieved by classifiers or threshold-based methods [1]. Popular approaches include finite state machines [21], [22], [23] and decision trees [24], [25], [26] for gait phase classification. However, trained models still tend to underperform when applied to new individuals or motions [27], highlighting challenges in adaptability and generalization across different individuals in automated pattern recognition through machine learning. Moreover, most classifiers or gait training approaches necessitate prior offline training [3]. Meanwhile, another significant challenge in real-time learning is data preprocessing, which includes data filtering. It is constrained by most forward (real-time) filters as they can only process data collected before the current sampling moment, resulting in the data lag in the processed data. One approach to address this problem is through predictive compensation, as in [28].

In addition to the aforementioned classification issues, the control law is a crucial component for the mid-level controller. One of the most commonly used methods is admittance control [3], which is proposed to mitigate the undesired dynamics imposed by LLEs. However, for classic admittance control, fixed parameters are insufficient to endow it with adaptability. Some researches therefore delve into the implementation of adaptive admittance control, such as [29], [30], and [31]. In real-time learning scenarios, there is a training phase where the exoskeleton needs to be compliant with the movement of the human lower limbs rather than providing assistance. This implies that the mid-level controller needs to adaptively balance this intervention on humans.

In addressing the challenge of transitioning between different gait patterns, this article focuses on motion gait within nested cycles, such as ascending stairs, aiming at real-time adaptability without relying on an explicit classifier. We integrate selective memory recursive least squares (SMRLSs) [32] with the radial basis function neural network (RBFNN) for supervised learning to develop a novel high-level controller, resulting in faster and improved transitions between different motion patterns compared to existing approaches. We tend to implement real-time learning to cope with unpredictable variations to enhance adaptability. Furthermore, equilibrium-based torque control (ETC) is proposed for the mid-level controller, using adaptive admittance control to achieve stable and adaptive assistance by controlling interaction torques.

The contributions of this article can be summarized as follows.

1) We adopt SMRLS for high-level controller to achieve faster real-time learning for assistance during the variation of motion, especially for scenarios within nested cycles.
2) The proposed real-time learning system allows filters to use both preceding and succeeding information of the current sample. Better filters before training can significantly improve performance, not only eliminating the delay problem caused by many real-time filters, but also optimizing the training samples to make the trained weights reliable.
3) To further transfer the prediction from the high-level controller into a balanced interaction, ETC with adaptive admittance control is proposed to modulate the magnitude of interaction torque intuitively and smoothly. It automatically balances the intervention on human limb between training and assistance phases, improving the LLE's adaptability.

The structure of this article is organized as follows. The scenario and system modeling are presented in Section II while Section III introduces the learning method of SMRLS-RBFNN as the high-level controller. Section IV provides the analysis of the system and the derivation of ETC as the mid-level controller. The results with analysis are depicted in Section V and Section VI by simulations and experiments respectively. Finally, Section VII concludes the contributions and discusses some limitations of this study.

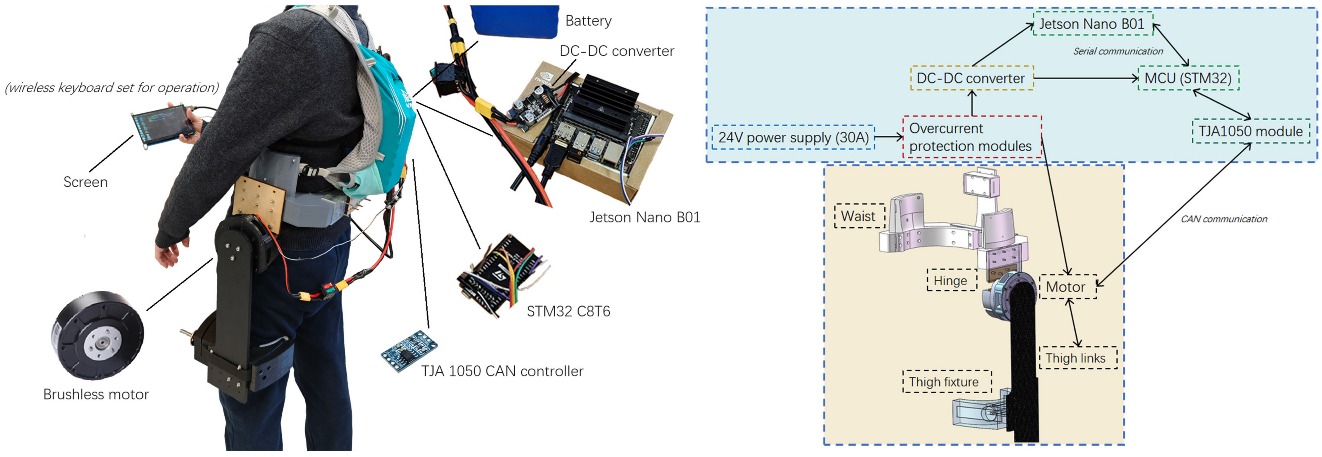
II. System Structure

The difficulties in designing exoskeletons arise from the necessity to optimize various aspects of the mechatronic system, such as its flexibility, weight, and power [33]. The proposed LLE system aims at assisting able-bodied individuals in walking more efficiently and effortlessly, with online learning and adaptation. To enable real-time walking assistance, a comprehensive LLE system is developed, incorporating dynamic modeling, hardware architecture, and the multilayer control strategy. The system features a high-level controller that operates independently of classifiers for human motion estimations, while the mid-level controller dynamically balances intervention in the human lower limbs between the stages of incomplete learning and assistance, converting the high-level controller's outputs into the appropriate assistance torque for the lower limbs.

A. Design of a System

1) Hardware Architecture: The structural configuration of the LLE system is illustrated in Fig. 1(a). In this article, a single-leg exoskeleton prototype is developed to evaluate the feasibility of the proposed control algorithm. The device incorporates two degrees of freedom (DOFs), specifically targeting at the hip joint. A hinge-like mechanism is implemented to facilitate abduction and adduction movements, while the design also enables thigh flexion and extension. This configuration provides the essential kinematic capabilities required to accommodate the complex motion patterns of the hip joint during locomotion.

The waist is composed of a circular structure that can rotate between 0° and 225°, with embedded straps for fixation. A



(a) (b)

Fig. 1. Structure of LLE system. (a) Components overview of wearable LLE. (b) Electromechanical strcutre with 3-D model.

portion of the LLE's weight is transferred to the shoulders by means of a wearable vest integrated into the system, which helps redistribute the load through the garment, enhancing user comfort. The complete exoskeleton, excluding power components, weighs approximately 2.42 kg with an extra weight of 1.94 kg of the battery and circuit (4.36 kg in total). Soft foam is added at contact positions around the waist, thighs, and back to enhance user comfort. Different screw holes are applied to the structure, allowing adjustments based on individuals. The waist fixation structure extends downward, connecting to the motor through hinges, ensuring that the center of the motor is fixed at the height corresponding to the hip joint. Consequently, the pivot of the motor aligns with the hip joint instead of the waist, causing a shift of the motor when receiving a significant torque. This shift might compromise assistance performance and influence data collection. To reinforce the stability of the structure, additional components have been added, extending from the waist upward to the back and integrated into the straps.

2) Electromechanical Structure: The motors employed in this study are sourced from the MIT Cheetah robot, featuring a built-in gear ratio of 9:1 and requiring 24 V power supply. The stall torque and nominal torque are 22 Nm and 11 Nm, with maximum and rated speeds of 230 rpm and 130 rpm, respectively. The motors then transmit data to the STM32 C8T6, which is regarded as the low-level controller, via CAN communication by TJA1050 module. Subsequently, the processed data is transferred to the Jetson Nano B01 for training through serial communication. The user carries a 24 V power supply battery in the backpack, with a maximum current of 30 A. Power is then provided directly or indirectly to microcontroller units (MCUs) and the actuator through overcurrent protection modules and a voltage DC-DC converter. The corresponding circuit structure can refer to Fig. 1(b).

B. Dynamic Model of Interaction System

In this article, we assume that there is a relative motion between the human leg and LLE (see Fig. 2). In the absence of external interactions, both of them ideally follow the feedforward dynamic model as

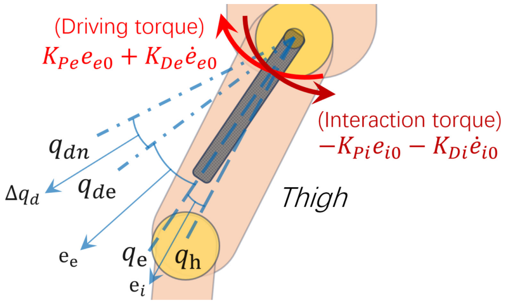


Fig. 2. Schematic of different angles and errors.

$$\tau_{h0} = M(q_h)\ddot{q}_h + C(q_h, \dot{q}_h) + G(q_h) \tag{1}$$

where M, C, G are, respectively, the inertia, Coriolis and centrifugal, gravitational matrices, q_h represents the rotation angle of human limb.

More specifically, the dynamic model of human can be regarded as a feedforward model with a PD feedback term which leads to the variation of the human limb motion

$$\tau_h = \tau_{h0} + K_{Ph}e_h + K_{Dh}\dot{e}_h \tag{2}$$

where e_h represents the tracking error between actual human limb angle and desired angle and τ_h is the torque exerted by human.

For human-exoskeleton interaction, we conceptualize it as a high-coefficient spring and damper connected between the human limb and the LLE, as below

$$\tau_l = \tau_{h0} + K_{Ph}e_h + K_{Dh}\dot{e}_h - \tau_i \tag{3}$$

$$\tau_i = K_{Pi}e_i + K_{Di}\dot{e}_i \tag{4}$$

$$e_i = q_e - q_h \tag{5}$$

where K_{Ph}, K_{Dh}, K_{Pi} and K_{Di} are coefficients of PD controller, τ_i is the interaction torque between the LLE and human limb (applied to the human limb from the LLE), q_e represents the angle of the LLE and τ_l is the torque exerted by human with the assistance of LLE.

Although q_e can be obtained accurately, the measurement of q_h is inaccurate as human limbs are soft and the gyroscope is difficult to adapt to the size of different people's limbs. Besides, e_i is small, and therefore the data accuracy is insufficient to meet subsequent computation requirements. Hence only q_e is available for the controller design, detected by the gyroscope or the inner encoder of motors.

The PD controller applied to the motor of the device follows:

$$\tau_e = K_{Pe}e_e + K_{De}\dot{e}_e + \tau_E \tag{6}$$

$$e_e = q_{de} - q_e \tag{7}$$

where K_{Pe} and K_{De} are the PD gains, e_e is the tracking error of the motor, q_{de} is the desired angle of the control objective and τ_E represents the offset torque of the LLE. Therefore, the control problem of the LLE becomes how to design the reference trajectory q_{de} of the LLE. A feasible solution is to make the trajectory of q_{de} ahead of the expected trajectory of human lower limb movement as

$$q_{de}(t) = q_h(t + t_\Delta) \tag{8}$$

where t_Δ is ahead time of the prediction.

In the learning stage where the neural network output is far from the expected trajectory, human needs to guide the motion of the entire system, and it is difficult to change the motion of the LLE only with (6). Therefore, admittance control is applied into the system as

$$D_A \Delta q_d + K_A \Delta \dot{q}_d = -\tau_i \tag{9}$$

where $\Delta q_d = q_{de} - q_{dn}$ is the altered angle according to admittance control, and q_{dn} is the output of predicted angle from the neural network. The inertia term of the admittance control is ignored in (9) considering the accuracy of the sensor and the derivation in the next section.

The whole system can be realized as two spring-damping models with admittance control, which is illustrated in Fig. 2.

Based on the above system and dynamic modeling, appropriate methods should be introduced into the mid-level and high-level controllers, which are discussed in details in the following two sections.

III. Real-Time Learning Based on SMRLS-RBFNN (High-Level Controller)

In order to improve the adaptability of real-time assistance, the high-level controller is required to quickly adapt to different trajectories with alternating cycles. Without the classifier, SMRLS-RBFNN is proposed to produce the reference trajectory by gait prediction.

A. Radial Basis Function Neural Network

The RBFNN, which has the capability of universal approximation, is a lightweight structure neural network with one hidden layer. According to [34], it is relatively less computationally demanding, so it is widely used in various applications [35], [36], [37].

In this article, the Gaussian function is used as the radial basis function

$$\phi_i(\chi) = \exp\left(\frac{-\|\chi - c_i\|^2}{2\sigma^2}\right) \tag{10}$$

where $i = 1, 2, \ldots, N$, N is the number of neurons, $\chi \in \mathbb{R}^l$ is the input vector, $c_i \in \mathbb{R}^l$ is the center of each radial basis function and $\sigma_i \in \mathbb{R}$ is the standard deviation which represents the receptive field width of $\phi_i(\chi)$.

The output of the RBFNN can be described as

$$f_{NN}(\chi) = \sum_{i=1}^{n} w_i\phi_i(\chi) = W^T\Phi(\chi) \tag{11}$$

where $f_{NN}(\chi) \in \mathbb{R}$ is the output, $W = [w_1, w_2, \ldots, w_N]^T \in \mathbb{R}^N$ is the weight vector, $\Phi(\chi) = [\phi_1(\chi), \phi_2(\chi), \ldots, \phi_N(\chi)] \in \mathbb{R}^N$ is the regressor vector composed of radial basis function $\phi_i(\chi)$.

For the scenario of this article, considering the learning from t_0 to t_1, the optimal weight vector is

$$W^* = \underset{W\in\mathbb{R}^N}{\arg\min}\left\{\int_{t_0}^{t_1} \|q_h(t) - W^T\Phi(t - t_\Delta)\|dt\right\}. \tag{12}$$

Common online learning methods used for RBFNN include stochastic gradient descent (SGD) and recursive least squares (RLSs). Although the RLS method uses the forgetting factor to enhance the sensitivity to new samples, a phenomenon of forgetting past data is inherent [32], as briefed in the following two sections.

B. Recursive Least Squares

The classic RLS method [38], [39] is usually implemented with forgetting factor λ as

$$\begin{aligned} W(k) &= W(k-1) \\ &\quad + P(k)\Phi(x(k))\big[y(k) - W^T(k-1)\Phi(x(k))\big] \\ P^{-1}(k) &= \lambda P^{-1}(k-1) + \Phi(x(k))\Phi^T(x(k)) \\ \Phi(x(k)) &= [\phi_1(x(k)), \phi_2(x(k)), \ldots, \phi_N(x(k))] \end{aligned} \tag{13}$$

where $P(k)$ is the covariance matrix at sampling time k, $\lambda \in (0, 1]$ is the forgetting factor during the recursive process and $x(k)$ and $y(k)$ represent the input sample and output sample, respectively.

C. Selective Memory Recursive Least Squares

Considering the scenario of real-time LLE assistance, it is necessary to rapidly and accurately obtain the expected gait prediction after the time of t_Δ. Nevertheless, the classic RLS in previous section will gradually forget the previous training sample in real-time learning because of the forgetting factor λ in (13). Even when $\lambda = 1$, the convergence will become slow as new samples are learned. Therefore, a derivative method of RLS named SMRLS is proposed [32]. Compared with RLS, it has the selective-memory term as the additional data input, leading to faster convergence and generalization ability.

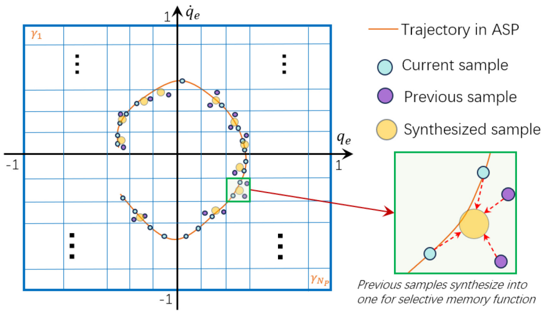


Fig. 3. Normalization and partitioning example of the the input space.

1) SMRLS-RBFNN:

$$\begin{aligned} W(k) = {} & W(k-1) \\ & + P(k)\Phi(x(k))\left[y(k) - W^T(k-1)\Phi(x(k))\right] \\ & - P(k)\Phi(\gamma_a(k-1))\Big[\varphi_a(k-1) \\ & - W^T(k-1)\Phi(\gamma_a(k-1))\Big] \\ P^{-1}(k) = {} & P^{-1}(k-1) + \Phi(x(k))\Phi^T(x(k)) \\ & - \Phi(\gamma_a(k-1))\Phi^T(\gamma_a(k-1)) \end{aligned} \tag{14}$$

where $\gamma_a(k-1)$ and $\varphi_a(k-1)$ represent the last synthesized input sample and output sample from the normalized range, respectively.

2) Normalization and Partitioning: The input range of SMRLS-RBFNN is normalized into a unit hypercube of the input space, such as $[-1, 1]$ for a 1-D input space. An example of input synthesis is shown in Fig. 3, where the trajectory is illustrated as the angle-speed plot phase (ASP). Current samples will then be partitioned into different lattices and synthesized as the new reference sample $(\gamma_a(k), \varphi_a(k))$ for the sample pairs in the corresponding lattice. The subscript $a = 1, 2, \ldots, N_P$ indicates the corresponding lattices of the current sample. The synthesis of $(\gamma_a(k), \varphi_a(k))$ can be accomplished through various methods. This article adopts a substitution approach that replaces $(\gamma_a(k), \varphi_a(k))$ by the normalized and partitioned sample from $(x(k), y(k))$ directly.

Generally, SMRLS can effectively alleviate the forgetting problem in classic RLS. The additional term will give balanced weight to samples in all lattices and works better in periodic variation of motion patterns.

D. Simultaneous Learning and Prediction

For the proposed method in this article, real-time learning is used for gait prediction in high-level controller, where they occur simultaneously. To generate the initial control objective, supervised learning based on SMRLS-RBFNN is conducted with both current and previous angle and angular velocity. Define n_a as the number of samples for the prediction and the ahead time of the prediction is therefore $n_a \Delta t$. For the LLE system in this article, $x(k)$ is represented by $q_e(k - n_a)$ and $\dot{q}_e(k - n_a)$, which are the angle and angular velocity of previous n_a moments, while $y(k)$ is represented by $q_e(k)$ and $\dot{q}_e(k)$. Therefore, at the current moment, the mapping between samples in previous moment and the current samples is trained in neural network as

$$f_{NN}(x(k)) = W^T(k)\Phi(x(k)) \tag{15}$$

$$W^T(k)\Phi(y(k)) = \begin{bmatrix} q_{dn} & \dot{q}_{dn} \end{bmatrix}^T \tag{16}$$

where f_{NN} is solved according to (12) and (14) for learning, the state $\begin{bmatrix} q_{dn} & \dot{q}_{dn} \end{bmatrix}^T$ is obtained by the prediction with neural network. Initial states of the angles, i.e., $q = 0$, correspond to the angles observed in the natural standing or hanging pose of the legs.

Meanwhile, real-time calculations using forward filters usually suffer from data lag issues. In the training of neural networks, even small fluctuations can cause significant training differences, as the entire training data set is affected. Moreover, the lag in data undoubtedly has a considerable impact on the assistance effect. Benefiting by the training approach involving forward prediction, both preceding and succeeding data of input samples can be collected and used in the filter in this article, leading to a better filtering effect on input samples.

The sliding window filter is used in this article. Define x_f as a $(2n_a + 1) \times 2$ matrix which includes the angles and angular velocities of $2n_a + 1$ previous samples. The final filtering result at the current time is $W^f x_f$, where W^f is the weight matrix and the weight of the central sample is $W^f_{(n_a+1)}$.

To ensure the weight distribution of the filter used is linearly proportional from edges to the center and the sum of them equals 1, the first element in weight matrix W^f_1 is defined as

$$W^f_1 = \frac{1 + W^f_{n_a+1}}{n_a + 1} - W^f_{n_a+1} \tag{17}$$

and W^f is positive, which means $W^f_{n_a+1} < (1/n_a)$.

Therefore, each element in W^f is defined as

$$W^f_i = \begin{cases} W^f_1 + (i-1)d_f, & i < n_a + 1 \\ W^f_{n_a+1} - (i - n_a - 1)d_f, & i \geq n_a + 1 \end{cases} \tag{18}$$

where $d_f = [(2n_a + 1)W^f_{n_a+1} - 1/(n_a + 1)]$ and W^f is settled once the central weight $W^f_{n_a+1}$ is defined.

Moreover, the sliding window filter integrates the information of a period of trajectory into each training sample.

IV. Mid-Level Controller of Real-Time Assistance

Even based on relatively accurate intention estimation, providing appropriate assistance remains a problem. The mid-level controller should balance the interaction torque during motion alternation and the assistance torque when the motion stabilizes. Thus, the ETC with adaptive admittance control is proposed in mid-level controller for adjustments of the generated trajectory based on SMRLS-RBFNN subsequently. The desired position and velocity are then sent to STM32 C8T6 as the low-level controller. The block diagram of the structure of the control system is illustrated in Fig. 4.

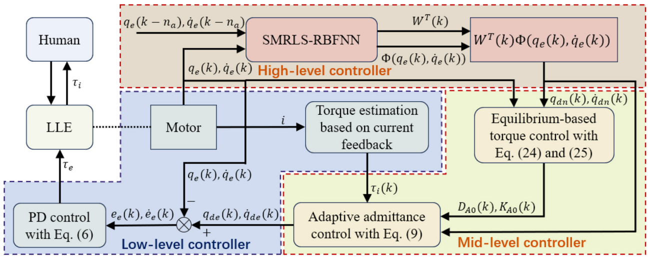

Human
τ_i
LLE
τ_e
PD control with Eq. (6)
$q_e(k-n_a),\dot{q}_e(k-n_a)$
SMRLS-RBFNN
$W^T(k)$
$W^T(k)\Phi(q_e(k),\dot{q}_e(k))$
$\Phi(q_e(k),\dot{q}_e(k))$
$q_e(k),\dot{q}_e(k)$
High-level controller
Motor
i
Torque estimation based on current feedback
$q_{dn}(k),\dot{q}_{dn}(k)$
Equilibrium-based torque control with Eq. (24) and (25)
$q_e(k),\dot{q}_e(k)$
$\tau_i(k)$
$-$
$e_e(k),\dot{e}_e(k)$
$q_{de}(k),\dot{q}_{de}(k)$
$+$
Adaptive admittance control with Eq. (9)
$D_{A0}(k),K_{A0}(k)$
Low-level controller
Mid-level controller

Fig. 4. Block diagram of the control system. The learning and prediction of angle and angular velocity are conducted in high-level controller simultaneously. The desired position and velocity are then generated through ETC in mid-level controller for low-level controller with PD control.

A. Equilibrium-Based Torque Control

After obtaining the prediction results through the RBFNN, the controller also needs to interpret q_{dn} and $\dot{q}_{dn}$ to achieve the desired assistance. In the early stages of learning or when human desires to change their current motion, the LLE should minimize excessive intervention on the human body. In the assistance stage, the LLE will assist human movement with τ_i. Thus, it is essential to ensure that the control laws of the mid-level controller can adapt to both stages.

Although defining a torque threshold can address this issue, the adjustment of the interaction torque needs to be continuous and smooth. According to the model in (3), (6), and (9), there exist equilibrium conditions that satisfy $\tau_i = \tau_e - \tau_E$ where the torque provided by the human is exactly offset by the resistance of LLE, resulting in a static balance. Therefore

$$K_{Pe}e_{e0} + K_{De}\dot{e}_{e0} = K_{Pi}e_{i0} + K_{Di}\dot{e}_{i0} = \tau_{i0} \tag{19}$$

where e_{e0}, e_{i0} and τ_{i0} are the particular coefficients and torque in the corresponding equilibrium conditions. Define $\Delta q_{ne} = q_{dn} - q_e$ as the initial ahead angle. During the operation, the direction of τ_{i0} is defined as $\operatorname{sgn}(\Delta q_{ne})$ with the magnitude of $|\tau_{i0}|$ as the motion direction varies.

B. Adaptive Admittance Control

The ahead angle Δq_{ne} varies in different moments with variable velocity. Moreover, when the legs are going to move oppositely, this difference inevitably decreases. Hence the training sample of the trajectory continuously changes with the invariant and relatively small coefficients in admittance control and human needs to retrain it continuously with effort. In this case, the variable parameters in admittance control based on (4), (6), and (9) are required.

To address the issue of interaction torque requirement in admittance control with an uncertain ahead angle, the same influence of interaction torque τ_{i0} on Δq_d with one certain equilibrium state is required. In this sense, τ_{i0} can be also regarded as a reference tolerance for interaction. Once the interaction torque τ_i reaches τ_{i0}, human limb and LLE will achieve the equilibrium condition.

Define ρ_{e0}, D_{A0}, K_{A0}, e_{e0}, and Δq_{d0} as the corresponding parameters in an equilibrium state. Without detecting e_i, once τ_{i0} is defined, we have

$$\begin{cases} \tau_{i0} = K_{Pe}e_{e0} + K_{De}\dot{e}_{e0} \\ -\tau_{i0} = D_{A0}\Delta q_{d0} + K_{A0}\Delta\dot{q}_{d0}. \end{cases} \tag{20}$$

By replacing e_{e0} and Δq_{d0} with $q_{\text{de}0} - q_e$ and $q_{\text{de}0} - q_{dn}$, there are still four unknown parameters $q_{\text{de}0}$, $\dot{q}_{\text{de}0}$, D_{A0} and K_{A0}. For easier computation, the internal parameters in (4), (6), and (9) can maintain the same proportion as

$$\begin{cases} \tau_{i0} = K_{Pe}(q_{\text{de}0} - q_e) + K_{De}(\dot{q}_{\text{de}0} - \dot{q}_e) \\ -\tau_{i0} = \rho_a K_{Pe}(q_{\text{de}0} - q_{dn}) + \rho_a K_{De}(\dot{q}_{\text{de}0} - \dot{q}_{dn}) \end{cases} \tag{21}$$

where ρ_a illustrates the same proportion with the human-LLE interaction model and $\rho_a K_{Pe} = D_A$, $\rho_a K_{De} = K_A$.

In this case, only ρ_a, $q_{\text{de}0}$, and $\dot{q}_{\text{de}0}$ are left unknown. To obtain D_{A0} and K_{A0}, only ρ_a is required instead of $q_{\text{de}0}$ and $\dot{q}_{\text{de}0}$. The first line in (21) can be written as

$$\tau_{i0} + K_{Pe}q_e + K_{De}\dot{q}_e = K_{Pe}q_{\text{de}0} + K_{Pe}\dot{q}_{\text{de}0} \tag{22}$$

where the left-hand side is completely known. Then we substitute (22) into the second line of (21)

$$\rho_a = \tau_{i0}(K_{Pe}q_{dn} + K_{De}\dot{q}_{dn} - K_{Pe}q_e - K_{De}\dot{q}_e - \tau_{i0})^{-1} \tag{23}$$

where τ_{i0}, K_{Pe} and K_{De} are predefined, q_{dn} and $\dot{q}_{dn}$ are output from the neural network and q_e and $\dot{q}_e$ can be obtained by motor drive. Replace $q_{dn} - q_e$ with Δq_{ne}, thus D_{A0} and K_{A0} are

$$D_{A0} = K_{Pe}\tau_{i0}(K_{Pe}\Delta q_{ne} + K_{De}\Delta\dot{q}_{ne} - \tau_{i0})^{-1} \tag{24}$$

$$K_{A0} = K_{De}\tau_{i0}(K_{Pe}\Delta q_{ne} + K_{De}\Delta\dot{q}_{ne} - \tau_{i0})^{-1}. \tag{25}$$

However, ρ_a may be equal to or even become negative when τ_{i0} exceeds $K_{Pe}\Delta q_{ne} + K_{De}\Delta\dot{q}_{ne}$ in some cases, causing D_{A0} and K_{A0} to become infinite or negative. If $K_{Pe}\Delta q_{ne} + K_{De}\Delta\dot{q}_{ne}$ is treated as a virtual torque τ_n against τ_{i0}, the boundary is defined as

$$\tau_n = \begin{cases} K_{Pe}\Delta q_{ne} + K_{De}\Delta\dot{q}_{ne}, & \tau_n > \rho_e\tau_{i0} \\ \rho_e\tau_{i0}, & \tau_n \leq \rho_e\tau_{i0} \end{cases} \tag{26}$$

where the coefficient $\rho_e \in (1, +\infty)$ defines the ratio of boundary between τ_n and τ_{i0}. In addition, to prevent equipment damage, it is necessary to establish a minimum value for ρ_a.

For the above system, K_{Pe} and K_{De} also limit the maximum value of D_A and K_A. Higher parameters will allow a larger range of variation in D_A and K_A. From a physical perspective, designing smaller motor control parameters K_{Pe} and K_{De} aims to provide less assistance, implying smaller interaction forces in the corresponding scenario. In such cases, it is logical to have smaller admittance parameters to facilitate easier trajectory variation through τ_i. Limiting their ranges by introducing ρ_a instead of D_A and K_A directly ensures that the range of variation in the admittance coefficient is self-adaptive, rather than requiring manual tuning by observing the experimental data.

V. Simulation

In the simulation, human movement is generated by (2). Two reference trajectories are defined as

$$q_r = \begin{cases} 62^\circ \frac{\pi}{180^\circ} \sin(2\pi t) & (n-1)T \leq t < 0.5nT \\ 18.6^\circ \frac{\pi}{180^\circ} \sin(2\pi t) & 0.5nT \leq t < nT \end{cases} \tag{27}$$

where $T = 25$ is the total period of the larger cycle and $n = 1, \ldots, N$. The motion changes between two types of

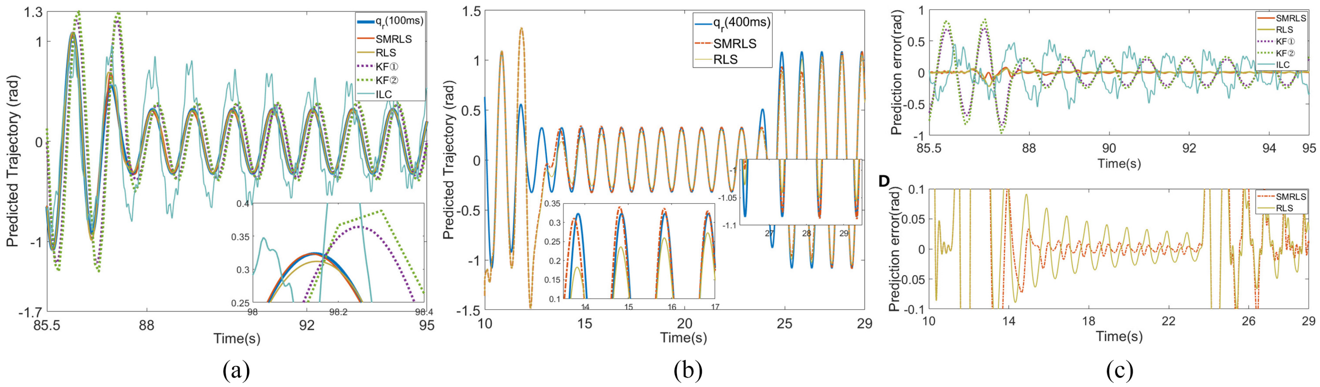


Fig. 5. Simulation results in open-loop system. To achieve the ahead prediction time as 100 ms, KF① reduces the sampling rate while KF② performs iterative prediction. (a) Prediction results when the ahead time is 100 ms. (b) Comparison of prediction results of SMRLS and RLS when the ahead time is 400 ms. (c) Prediction error in (a). (d) Prediction error in (b).

trajectory in q_r, which simulates the scenarios of constantly ascending stairs. According to [40], the range of human gait frequency (per minute) during walking is generally 52–82 Hz. The dominant frequency of both trajectories is 60 Hz which is consistent with it. The sliding window filter will not be implemented into simulation as the data is generated ideally.

The other parameters are set as: the leg is regarded as a mass point with $m_h = 12$ kg, while the mass of LLE is regarded as $m_e = 0.5$ kg. The length of leg $l = 0.5$ m (between center pivot and the mass point).

The other control parameters are set as follows: $K_{Ph} = 200$, $K_{Dh} = 50$, $K_{Pe} = 32$, $K_{De} = 8$, $K_{Pi} = 100$, and $K_{Di} = 25$. The angular acceleration is calculated from angular velocity by transfer function as $(1.02s/[0.2s + 1])$.

For RBFNN settings, the number of nodes is 25 (5×5 for angle and velocity respectively) and $\sigma = 0.35$ after normalization to $[-1, 1]$. The data of angle and angular velocity are sampled into 100 points, respectively, in SMRLS, which means it is divided into 10000 lattices for previous recording. To avoid negative effects caused by the neural network during early training, it only produces output to the system when $t > 10$ s. The forgetting factor $\lambda = 1$ for RLS.

The following prediction methods for the trajectory are considered for comparison.

Method-1: The proposed method with SMRLS-RBFNN.

Method-2: RLS-RBFNN.

Method-3: Model prediction with Kalman filter (KF), where the measurement matrix is $\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$, the transition matrix is $\begin{bmatrix} 1 & 5\Delta t \\ 0 & 1 \end{bmatrix}$, and the state vector is $\begin{bmatrix} q \\ \dot{q} \end{bmatrix}$.

Method-4: ILC [41]. Each period of ILC is set as 1 s and the sample frequency is set as 50 Hz (20 ms).

A. Open-Loop System

This section shows the training results without interaction feedback to the control system, which compares results of different methods for human intention prediction in ideal cases.

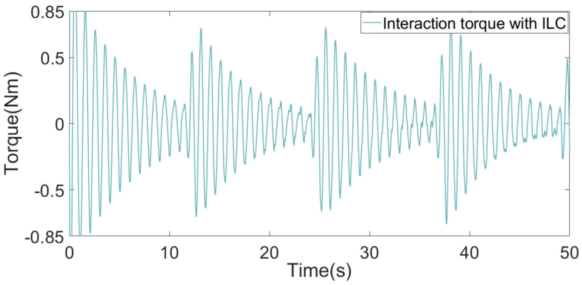


Fig. 6. Interaction torque with ILC method.

As shown in Fig. 5(a), ILC cannot converge appropriately with nested periodic variations even though this kind of variation is very common in ordinary motion. As the learning gain is relatively high, fluctuations can be clearly observed even with a 2-order sliding window filter. Fig. 6 further indicates that the interaction torque is gradually converging, but the trajectory enters the next subperiod before it fully converges, so it does not work successfully. Although [42] proposed a method to improve the adaptability of ILC in lower-limb assistance, it is subject to the same problem of ILC in nested period scenarios, especially when the variation is rapid and regular.

For the fixed model with KF, two different approaches are attempted. The first one involves maintaining the same sampling period as other methods (20 ms) but iterating predictions for five times to achieve the same ahead time (100 ms). The second approach involves setting the sampling period exactly equal to the lead time. It is obvious that both approaches cannot predict the trajectory well. This is partially due to the lag problem caused by KF itself. Besides, due to the limitation of the fixed linear model, it lags when the speed is fast and overestimates at low speed. The tracking errors are shown in Fig. 5(c), which shows the large error of the rest method compared with RLS and SMRLS.

Although SMRLS is better than RLS in ahead time of 100 ms, the difference is not significant in lower ahead time (100 ms). Hence Fig. 5(b) illustrates the difference in higher ahead time (400 ms). During the first transition (13.5–17 s), SMRLS converges to the reference faster, though the sample of this motion is new to both neural training methods, which shows better generalization of our proposed method. During the subsequent first return of motion (26–29 s), although RLS method converges slightly better in the first wave peak, it consistently fails to fully converge to the reference trajectory and exhibits significantly slower convergence in subsequent iterations. The tracking errors of both methods are illustrated in Fig. 5(d), showing that SRMLS performs better than RLS with larger ahead of time.

VI. Experiment

In this section, only RLS will be used for comparison as the other prediction methods obviously show poor usability in the simulation that might damage the LLE device. Similar with the simulation, the neural network provides output after 10 s of training while the actuator will not assist until 12 s. The sampling, communication and computation frequencies below are all set as 66.7 Hz ($\Delta t = 15$ ms). The data are sampled into 100 points and the number of nodes in RBFNN is 25 (5×5) and $\sigma = 0.4$ after normalization to $[-1, 1]$. In order to demonstrate adaptability of the proposed method, all datasets in experiments do not undergo pretraining, which means the initial weight vectors W of RBFNN are all set as zero. The forgetting factor $\lambda = 1$ for RLS. The normalization range of angle is set as $[-0.2, 1.2]$ while the normalization range of angular velocity is set as $[-2, 2]$. The data in the experiment are collected from a human subject who has no knowledge about the research conducted. In the initialization of program, the subject is asked to maintain a natural standing posture, with their legs perpendicular to the ground. The angle of motor drive is then calibrated to 0.

1) Scenario Setting: This experiment primarily focuses on the scenario of ascending stairs. Typically, humans do not maintain a constant stair-climbing state but alternate between climbing stairs (CS stage) and level walking (LW stage) after ascending a flight of stairs. This movement involves a nested periodic variation of motion patterns, requiring a continuous transition of lower limb motion.

For evaluation purposes, the stairs in this experiment are emulated by a single step with a height of 17.5 cm as shown in Fig. 7. As the experiment predominantly focuses on the movement of one leg, the subject will repeatedly ascend this step using their left legs, mimicking the downward stepping motion of one leg during CS stages, and then shift their center of gravity to the right leg while bending the left leg to simulate the lifting motion when ascending stairs. Throughout this process, the contact position between the left leg and the stair will remain constant. After completing several cycles of these actions, the subject will move away from the step and walk on flat ground for a short distance, and then return to the step again, commencing the next cycle.

A. Open-Loop System

This section shows the offline training result with SMRLS and RLS, where the training samples are collected from one of the closed-loop experiments. Compared with the simulation, even within the same stage, samples include disturbance and noise, which requires better generalization and faster adaptability. However, the normalization range of RBFNN is set to fit in with the input sample range as offline training, which is different from the closed-loop experiment. The number of samples for ahead training n_a is 8, which means the ahead time of prediction is 120 ms.

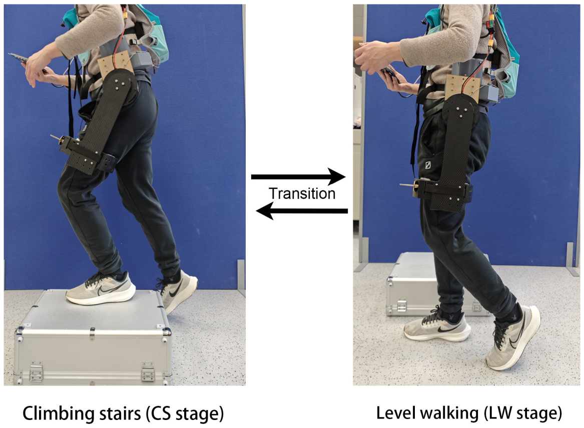
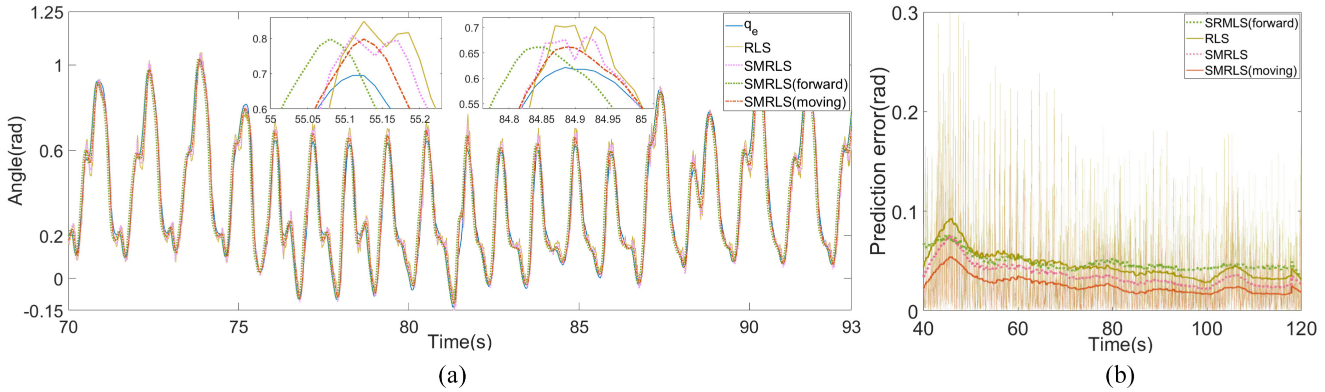


Fig. 7. Two motion cycles in the scenario of ascending stairs. The left figure shows the first stage of stair climbing and the right figure shows another stage of level walking, while the motion of subject switches between two patterns. The motion of CS stage involves a single step, simulating the action of ascending stairs by stepping up and down.

The comparison of the forward filter and sliding window filter is shown in Fig. 8. During this experiment, the subject initially performed the stair-climbing action until the system reached a relatively stable state before transitioning to an alternative motion. The approximate time intervals for the CS stages were: 58–75 s and 87–103 s, while the intervals for LW stages were: 75–87 s. According to Fig. 8(a), evidently, although the convergence of all methods after switching motions gradually improves with the training, the performance of RLS is poor. In the initial transition, SMRLS's superior accuracy in prediction can be attributed to its better generalization. However, during the first time of switching back, a noticeable gap between the two methods is observed, indicating that our method benefits from its selective-memory capability. For the derivative methods, a significant phase difference occurs using the forward filter. The forward filter introduces data lags, and when used for training input samples, it effectively increases the time interval between input and output samples. Consequently, when using the trained mapping, it leads to a more advanced phase of output from the corresponding input. The prediction error might suddenly increase after the transition of motion, but SMRLS can rapidly reduce the prediction error back to a lower magnitude according to Fig. 8(b). The average error after 20 s of program execution is shown by Table I. SMRLS method with forward filter even performs worse than RLS mainly due to its phase difference. According to results, the proposed filter not only reduces fluctuations, but also reduces prediction error as these subtle fluctuations lead to poorer training samples, thus affecting the overall training effectiveness of the entire dataset, in the sense of convergence and generalization.

B. Assistance in Climbing Stairs (Closed-Loop System)

This section will demonstrate the overall assistance performance of the entire system. Merits and drawbacks of methods will be mainly manifested through interaction



(a)

(b)

Fig. 8. Comparison of experiment in open-loop system. (a) Predicted angle when the ahead time is 120 ms. (b) Prediction error where translucent curves show the raw data while opaque curves show the data processed by filter.

TABLE I
MEAN OF PREDICTION ERROR

	Moving	SMRLS	Forward	RLS
$\bar{E}_{err}$	0.0221	0.0320	0.0428	0.0407

torques. The derived ETC will also be compared with fixed-parameter admittance control and PD control. Experiments are conducted using varied impedance parameters in PD control, including both larger and smaller settings. This was employed to elucidate results across diverse magnitude requirements for assistance torque, thereby offering insights into the system's performance under different conditions.

1) Settings: The number of samples for ahead training n_a is 12, which means the ahead time of prediction is 180 ms. As mentioned in the description of open-loop experiment, the first switch in motion mode occurs when the subject perceives that it is relatively stable. For the limitations of ETC in all experiments, $\rho_e = 1.01$ and $\rho_{\min} = 0.4$.

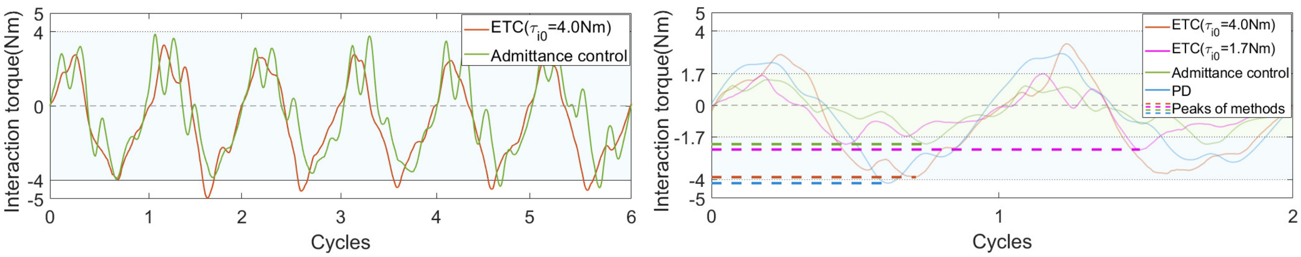
When lowering the leg at a slow speed, the proposed method will fail to provide assistance but introduce resistance. However, this issue tends to be mitigated in the stair-climbing scenario. Therefore, in this experiment, the distance when the leg is lowered to make contact with the step is disregarded, assuming that a person immediately makes contact with the step upon lowering the leg. For the ideal assistance of ordinary walking (LW stage), the interaction force should be backward only when in contact with the ground as well, and offering forward assistance during other phases. However, there is relatively less contact with the ground during LW stages, which means the movement of legs suspended and lowering cannot be ignored.

The magnitude of torque for assistance is mainly restricted by K_{pe}, K_{de} and the ahead time of prediction. In this experiment, the performances of ETC, admittance control and PD control policy in relatively higher assistance parameters are investigated, where $K_{pe} = 45$, $K_{De} = 5$ and $\tau_{i0} = 4.0$ Nm. For admittance control, $D_A = 20$ and $K_A = 2.5$, while PD control can be interpreted as admittance control with infinite parameters. For the relatively smaller assistance, $K_{Pe} = 20$, $K_{De} = 3$. The parameters of admittance control remain the same as in high assistance conditions ($D_A = 20$ and $K_A = 2.5$ for admittance control). The equilibrium torque of ETC is set as $\tau_{i0} = 4.0$ Nm and $\tau_{i0} = 1.7$ Nm in two conditions, which represent the maximum desired assistance torque of 4.0 Nm and 1.7 Nm, respectively.

2) Interaction Torque Comparison With Control Laws: To highlight the contribution related to the control law in this article, the results using ETC, admittance control and PD control in CS stages are shown in Fig. 9. For clear demonstration, curves of torque are scaled in time domain to normalize different cycles.

Fig. 9(a) shows the results using the ETC and admittance control in higher assistance condition. It is clear that ETC provides smoother result which indicates more stable assistance. This is also related to its better adaptation. For example, the fixed parameters of admittance control used in this article are relatively low in such a requirement of larger assistance that little force will cause larger Δq_d (variation of q_{de}), which easily leads to more fluctuations. Due to the learning of RBFNN, those poor samples of fluctuation will be sent back for learning again as a vicious circle. In addition, the results of the current trajectory q_e, the predicted trajectory from RBFNN q_{dn} and the interaction torque τ_i using the classic PD control are presented in Fig. 10, which indicates that PD control cannot even work successfully in this condition. Higher parameters in PD control represent larger upper-limit of assistance torque, which however needs admittance control for movement compliance. Although the training in RBFNN is generally completed within 12 s and it provides acceptable assistance within two initial steps (12–14 s), the movements can be also different even in the same motion stage, which causes poor and rigid assistance.

Admittance control with these fixed parameters and PD control works in lower assistance requirement as shown in Fig. 9(b), which also illustrates the results of ETC with $\tau_{i0} = 4.0$ Nm and $\tau_{i0} = 1.7$ Nm respectively. However, without tuning of other parameters, the magnitude of interaction torque using admittance control decreases as PD parameters decrease, while ETC with $\tau_{i0} = 4.0$ Nm still generally maintains the interaction torque in 4.0 Nm as before. Although sligtly smaller than that in Fig. 9(a), PD control also restricts the magnitude of torque within 4.0 Nm as the upper-limit.



(a)

(b)

Fig. 9. Results of different control methods, where the parameters $\tau_{i0} = 4$ Nm and $\tau_{i0} = 1.7$ Nm of ETC are highlighted by blue and green background regions respectively (Curves are scaled horizontally to match different cycles). (a) Results of ETC and classic admittance control in larger impedance setting. (b) Interaction torque results of ETC (with two different settings), admittance control and PD control in lower impedance setting.

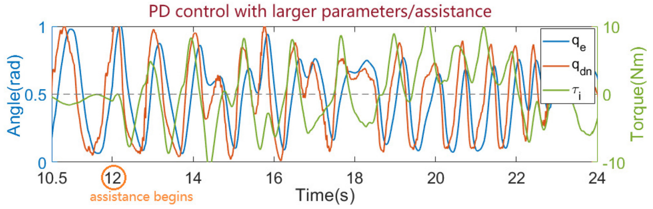


Fig. 10. Poor and disordered assistance using PD control in larger impedance setting.

Meanwhile, ETC with $\tau_{i0} = 1.7$ Nm generally restricts the interaction torque in 1.7 Nm as a comparison, which means it is easier and more intuitive to adjust the assistance torque with ETC and can better adapt to changes caused by the rest settings.

3) Comparison Between SMRLS and RLS: The results with SMRLS and RLS both tend to achieve a relatively stable state after a period of training, as shown in the simulation with Fig. 5(a), with fast convergence. Although the prediction error with RLS is larger after convergence, this subtle difference is not apparent in experiments. Therefore, the difference between SMRLS and RLS is mainly reflected in the early stage where the learning is not complete.

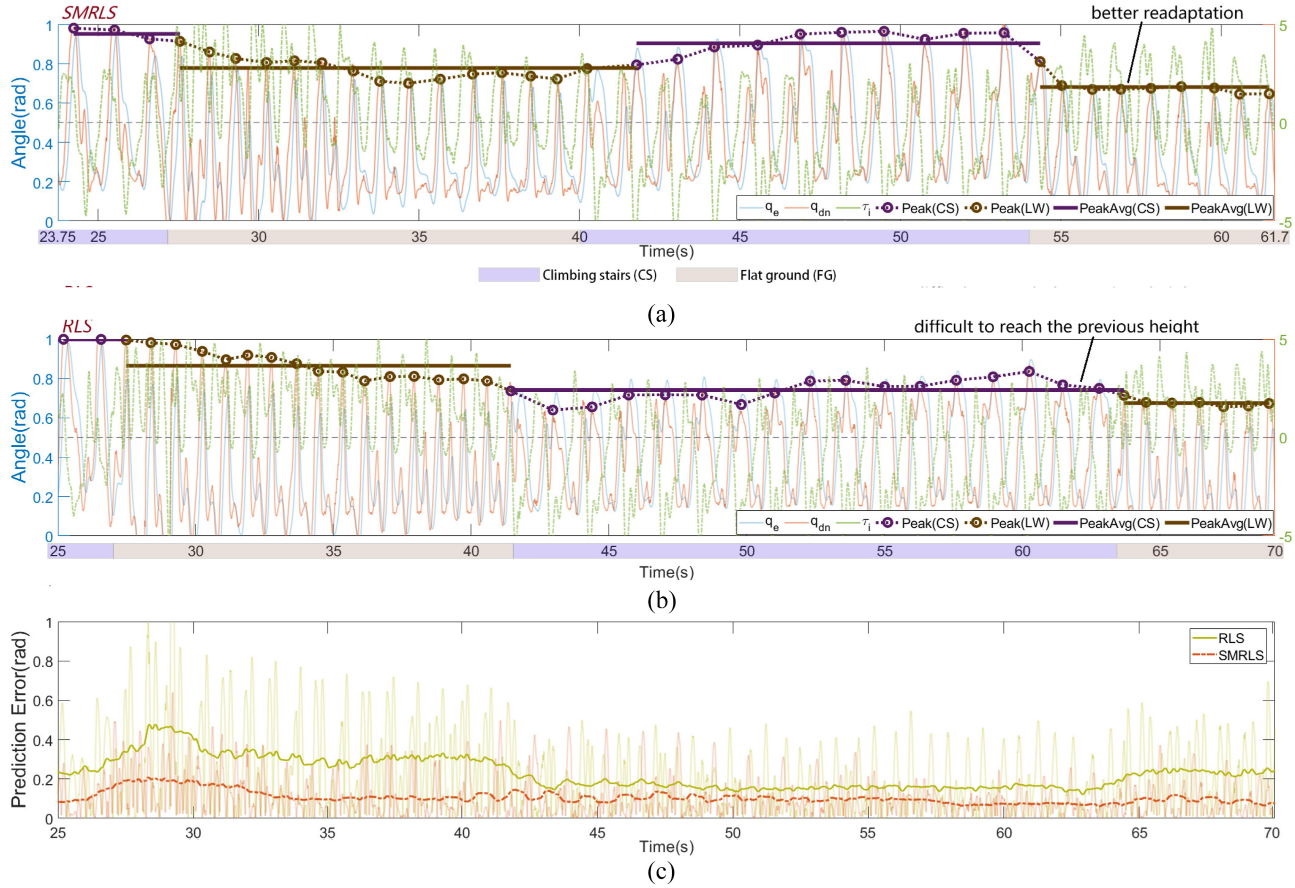
In Fig. 11, the examples of segment with SMRLS and RLS from the first transition into the LW stage to the second transition into it are depicted. The corresponding prediction error, which is processed by the sliding window filter with a window size of 150, is presented in Fig. 11(c). The prediction error of RLS in experiment is obviously larger compared with the results shown in the simulation. Moreover, the prediction error with RLS clearly increases during the transition of different stages, while the transition cannot even be distinguished from the result of SMRLS and the average prediction error remains within 0.1rad(6°) after the second cycle.

SMRLS settles the assistance after the first six steps during the first transition (after 33 s) and the interaction torque begins to provide assistance, while the results with RLS still cannot settle even when it approaches the first return to the initial motion pattern. The overall direction of assistance leans toward the forward direction as the prediction still reaches higher angle, resembling the motion of CS instead of LW. Similarly, during the first transition back to CS stage, it fails to converge effectively, while SMRLS generally converges after three steps (after 44 s). Similar results also can be found in second transition into the LW stage.

In addition, when using RLS for the second transition into CS stage, a noticeable reduction in the maximum amplitude of the subject's leg lift can be observed compared with results using SMRLS, which means this method affects the subject's native motion rather than providing assistance in turn. In this case, the subject will subconsciously lean the torso backward during walking according to our experiments. It is also clear that the subject tried to resist this poor assistance and exerted additional effort to correct this error, preventing it from persistently affecting subsequent training as flawed samples. It however results in an overall bias of assistance toward the backwards direction, causing the assistance to respond more rapidly with negative torque during the leg-lifting phase as the resistance, similar to the drawback of admittance control. It will produce worse assistance if the subject tended to fix it. In other words, the subject must either tolerate the inappropriate alterations imposed by the LLE or exert additional effort to correct this issue.

C. Assistance Rate

In this experiment, the assistance rate (AR) is introduced to quantify the proportion of correctly assisted movements using different control methods. If the direction of the interaction torque aligns with the corresponding theoretical direction mentioned above, it is considered beneficial assistance. Therefore, the correct assistance can be defined as follows, in the context of CS stages: 0%–100% for the leg lifting stage (positive), 0%–100% for the leg lowering stage (negative). For LW stages: 0-20% and 85%–100% for the backward swing phase and 0%–100% for the forward swing phase (positive), and 20%–100% for the backward swing phase (negative). Hence 20% is set as the standard in this article. However, a certain phase difference of 10% is allowed between the interaction torque and the ideal point. When the phase is leading, it can be regarded as the exoskeleton providing a torque resisting inertial effects and altering the current state. When the phase is lagging, it can be regarded as the exoskeleton following the user's intent and providing assistance. Both forms contribute successfully to providing assistance. Due to significant uncertainty and disturbances in the data, segmentation of each stage will be marked manually. According to [43], the stance phase (where the foot is in contact with the ground) averagely



(a)

(b)

(c)

Fig. 11. Example of results of SMRLS and RLS with ETC. Each figure depicts results from the first transition to the LW stage to the second transition. The CS and LW stages are highlighted by different colors, while peak trends are illustrated respectively as well. (a) Results using SMRLS. (b) Results using RLS. (c) Prediction error of SMRLS and RLS in example segments, where translucent curves show the raw data while opaque curves show the data processed by filter.

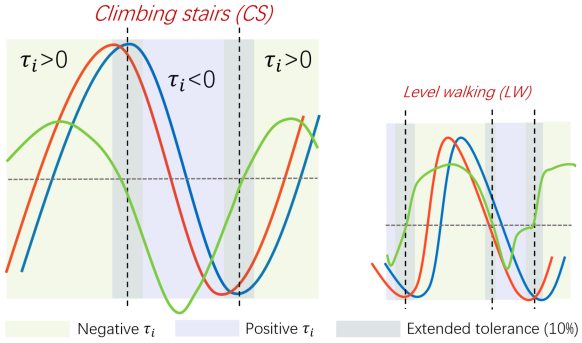


Fig. 12. Ideal assistance for AR used in this article.

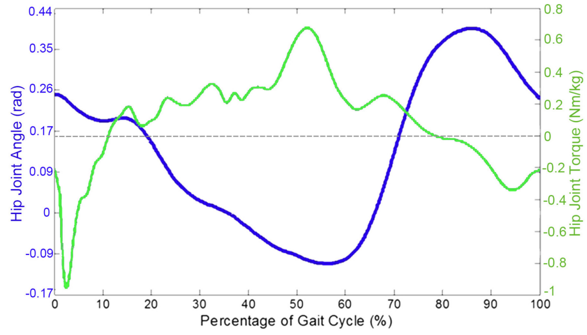
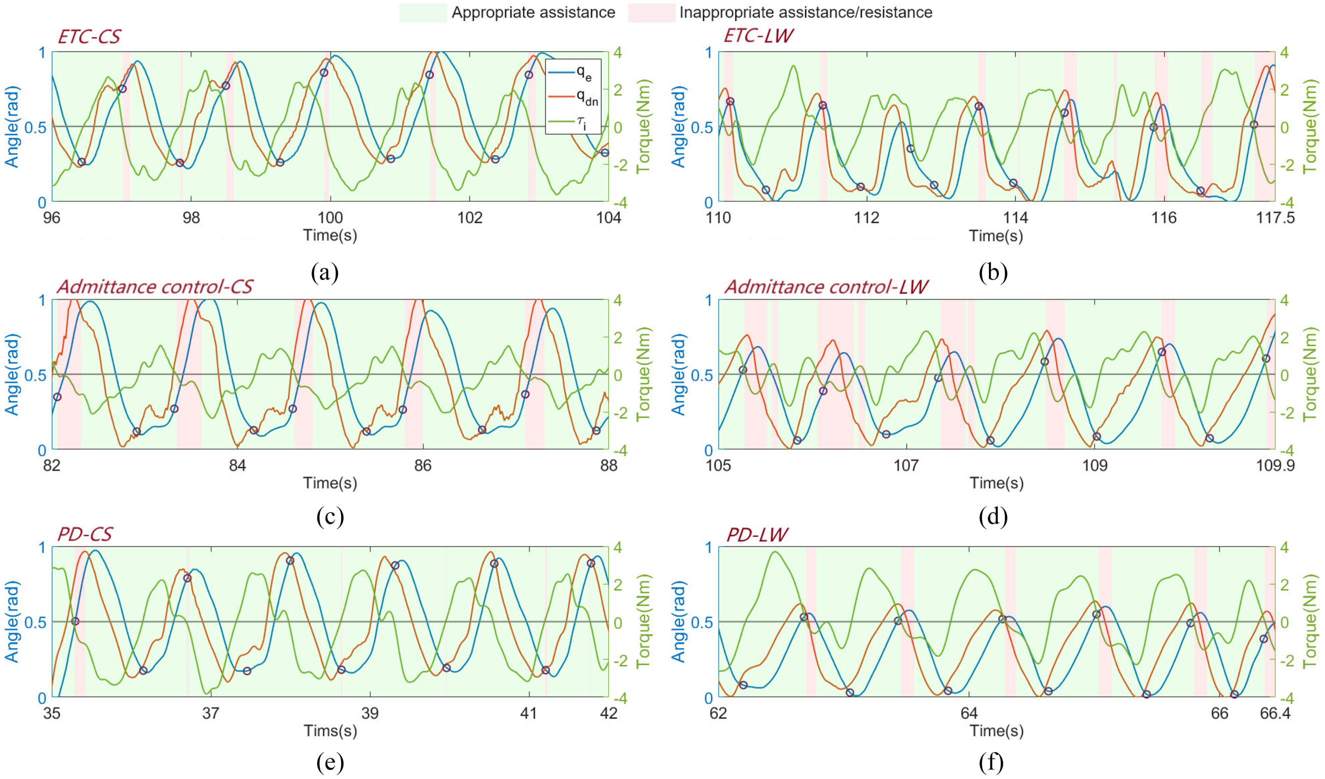


Fig. 13. Example relationship between human hip joint angles and torques during a gait cycle of the LW stage from [43] and [44], where the unit of Nm/kg indicates the exerted torque divided by the body mass.

occupies 62% (time) of a single gait cycle. Hence, more torque should be directed forward during ordinary walking. In this circumstance, the ideal assistance should resemble that depicted in Fig. 12.

Fig. 13, which is from [43] and [44], presents a data example of hip joint angles and torques within one gait cycle during normal human walking (LW stage) according to the clinical gait analysis. Although there exists a slight discrepancy between the definition of AR and the provided example, the curve trend of the defined AR and curves of experiment results presented below are generally consistent with it, which confirms the rationality of comparisons in this article.

An example of how AR is calculated in weaker assistance condition is presented in Fig. 14, where the moment of reversal of the interaction torque has been marked on the respective trajectory. Evidently, ETC provides more segments of appropriate assistance (green segments) according to AR compared with admittance control as the interaction torque with the admittance control decreases earlier while legs are



(a) (b) (c) (d) (e) (f)

Fig. 14. Comparison of different control methods in lower assistance parameters. The appropriateness of assistance is judged by the defined AR. (a) ETC-CS. (b) ETC-LW. (c) Admittance control-CS. (d) Admittance control-LW. (e) $PD - CS$. (f) $PD - LW$.

still lifting. The desired position and velocity q_{de} and $\dot{q}_{de}$ are easily changed by a smaller torque as the ahead angle and angular velocity of prediction decrease with invariant parameters in admittance control, even though this magnitude of torque is necessary for assistance. Although PD control has better AR, it cannot even provide assistance in a larger assistance condition as shown in Fig. 10, which means the good performance of the PD control is occasional. In addition, low PD parameters lead to low upper-limit of assistance torque, while the torque for assistance usually requires a greater margin for the other control methods. For example, it already restricts the maximum assistance torque of ETC in such condition though ETC still performs well.

Eventually, AR of different experiments above is summarized in Fig. 15, which is calculated from 30 s after the operation. It shows that ETC produces more proper assistance compared with admittance control. Although PD controller performs well in lower assistance, it cannot work in higher assistance, which is thus not included in Fig. 15.

For ordinary walking, the movement of human legs is largely influenced by inertia and gravity. Therefore, a significant portion of human effort is dedicated to maintaining standing balance [44]. This is also achieved through the knee and ankle joints. According to the subject, they experience significant assistance when support is provided during the intermediate phase of forward swing and the initial moments of ground contact. In other stages, even if the assistance direction differs from the AR definition above, the subject does not experience a noticeable impact as long as the resistance is not very large.

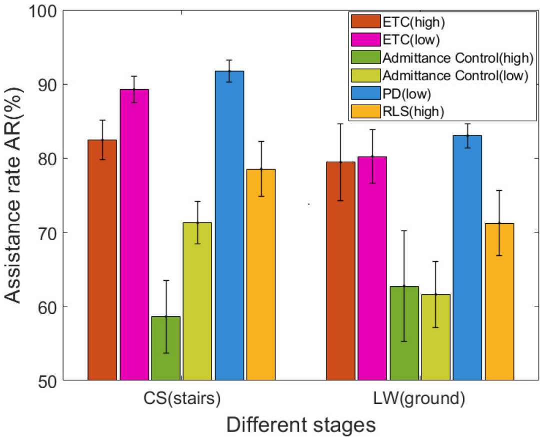


Fig. 15. AR in experiments with different methods, where (high) and (low) represent the experiment with different assistance levels respectively. ETC, admittance control and PD all use SMRLS except RLS. The data of PD (low) is not displayed as it cannot work.

AR in this article only considers the direction of torque without the magnitude. For instance, in Fig. 9(a), the interaction with admittance control is apparently unstable, but it will not show much difference in AR. Therefore, AR generally proves that ETC and SMRLS are better than the other methods, but there are still many merits which cannot be intuitively observed through the current results.

VII. Conclusion and Discussion

This article proposes a walking assistance strategy with real-time learning based on SMRLS-RBFNN. A control policy ETC is derived to balance the interaction torque using the prediction of trajectory in order to achieve more stable and controllable assistance. In contrast to many existing methods that rely on offline learning, such as sim2real [13], the strategy proposed does not use classifiers for high-level control. Instead, it achieves enhanced adaptability through optimization of neural network training methods. Moreover, compared to many complex real-time learning methods [16], the setting and neural network structure involved in this strategy are relatively simple. This simplicity significantly reduces data processing time, thereby greatly enhancing control and communication frequencies which are crucial factors for real-time learning and assistance. According to further tests, SMRLS-RBFNN can not only operate in the standard 5V/4A power supply mode of Jetson Nano, but also in a low-power mode of 5 V/2 A.

In simulations, comparisons are made with other trajectory prediction methods which do not rely on neural networks. These include methods based on a fixed model with external optimization (KF) and methods involving continuous training with iterative and recursive properties (ILC). The proposed strategy in this article demonstrates superior adaptability compared with other methods in addressing specific scenarios. Comparisons are also made with the classical RLS method in both simulation and experiment. The results confirm that SMRLS-RBFNN demonstrates enhanced adaptability to previously trained data and superior generalization. It converges more rapidly with variations in the motion period. Although this article focuses on the scenario of ascending stairs to highlight the adaptability of nested cycles and better convergence during real-time learning, the proposed approach is not limited to this context. Although the gait ahead prediction method may not be applicable to scenarios requiring individuals to perform negative work, such as descending stairs, where resistance against leg weight or maintaining balance is necessary, it can still facilitate an increase in the human's step frequency as augmentation. The definition of AR is based on the assumption of ideal situations, but the experiment results are generally consistent with results from the gait analysis in the literature, such as [43] and [44].

Although the assistance torque in the experiment is limited within 5 Nm by ETC parameter τ_{i0}, the proposed method can provide assistance far exceeding this value. It is worth noting that the required assistance torque can exceed the suitable interaction torque when switching the motion pattern. However, ETC with the fixed parameter τ_{i0}, used as a threshold for switching of motion patterns, also limits maximum assistance torque. One solution is to set the upper and lower bounds of τ_{i0} and adjust it based on motion trajectory stability, which can be a potential direction for future research. For example, as the human motion trajectory stabilizes, τ_{i0} can gradually increase to its upper-limit. Although this method may also work in different control laws compared in this article for balancing interventions from the LLE, ETC's advantages still remain, including intuitive parameter tuning and stable assistance. Moreover, as the threshold, fluctuations in τ_{i0} during the adaptive process are unlikely to significantly affect assistance, unlike direct adjustments to the admittance coefficient or PD parameters, which may lead to instability.

In addition, the proposed method can also be used in other application scenarios of human-robot interaction, such as upper limb movement assistance or collaborative manipulators. Actually, the ILC method that is used for comparison in the simulation section has been applied to corresponding scenarios in [41]. The proposed method in this article particularly works for real-time learning with periodic variations, thus rehabilitation can also be a potential scenario [45].

References

[1] R. Baud, A. R. Manzoori, A. Ijspeert, and M. Bouri, "Review of control strategies for lower-limb exoskeletons to assist gait," *J. Neuroeng. Rehabil.*, vol. 18, no. 1, p. 119, 2021.

[2] J. L. Pons, "Rehabilitation exoskeletal robotics," *IEEE Eng. Med. Biol. Mag.*, vol. 29, no. 3, pp. 57–63, May 2010.

[3] M. R. Tucker et al., "Control strategies for active lower extremity prosthetics and orthotics: A review," *J. Neuroeng. Rehabil.*, vol. 12, no. 1, pp. 1–30, 2015.

[4] B. Chen et al., "Recent developments and challenges of lower extremity exoskeletons," *J. Orthopaedic Transl.*, vol. 5, pp. 26–37, Apr. 2016.

[5] L.-F. Yeung et al., "Design of an exoskeleton ankle robot for robot-assisted gait training of stroke patients," in *Proc. Int. Conf. Rehabil. Robot. (ICORR)*, 2017, pp. 211–215.

[6] V. R. Garate et al., "Walking assistance using artificial primitives: A novel bioinspired framework using motor primitives for locomotion assistance through a wearable cooperative exoskeleton," *IEEE Robot. Autom. Mag.*, vol. 23, no. 1, pp. 83–95, Apr. 2016.

[7] J. Jang, K. Kim, J. Lee, B. Lim, and Y. Shim, "Online gait task recognition algorithm for hip exoskeleton," in *Proc. Int. Conf. Intell. Robots Syst. (IROS)*, 2015, pp. 5327–5332.

[8] F. Wang, L. Yan, and J. Xiao, "Human gait recognition system based on support vector machine algorithm and using wearable sensors," *Sensors Mater.*, vol. 31, p. 31, Apr. 2019.

[9] N. Aphiratsakun and M. Parnichkun, "Balancing control of AIT leg exoskeleton using ZMP based FLC," *Int. J. Adv. Robot. Syst.*, vol. 6, no. 4, p. 34, 2009.

[10] T. Gurriet et al., "Towards restoring locomotion for paraplegics: Realizing dynamically stable walking on exoskeletons," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, 2018, pp. 2804–2811.

[11] R. Huang, Q. Wu, J. Qiu, H. Cheng, Q. Chen, and Z. Peng, "Adaptive gait planning with dynamic movement primitives for walking assistance lower exoskeleton in uphill slopes," *Sensors Mater.*, vol. 32, no. 4, pp. 1279–1291, 2020.

[12] T. Gurriet, M. Tucker, A. Duburcq, G. Boeris, and A. D. Ames, "Towards variable assistance for lower body exoskeletons," *IEEE Robot. Autom. Lett.*, vol. 5, no. 1, pp. 266–273, Jan. 2020.

[13] S. Luo et al., "Experiment-free exoskeleton assistance via learning in simulation," *Nature*, vol. 630, no. 8016, pp. 353–359, 2024.

[14] L. Righetti, J. Buchli, and A. J. Ijspeert, "Adaptive frequency oscillators and applications," *Open Cybern. Syst. J.*, vol. 3, no. 1, pp. 64–69, 2009.

[15] D. A. Bristow, M. Tharayil, and A. G. Alleyne, "A survey of iterative learning control," *IEEE Control Syst. Mag.*, vol. 26, no. 3, pp. 96–114, May 2006.

[16] Y. Hua et al., "A novel weight-bearing lower limb exoskeleton based on motion intention prediction and locomotion state identification," *IEEE Access*, vol. 7, pp. 37620–37638, 2019.

[17] T. Igami, Y. Akiyama, S. Okamoto, and Y. Yamada, "Development of adaptive gait assist algorithm using ground reaction force," in *Proc. Workshop Robot Control (WROCO)*, Daejeon, South Korea, 2019, pp. 1–4.

[18] S. Au, M. Berniker, and H. Herr, "Powered ankle-foot prosthesis to assist level-ground and stair-descent gaits," *Neural Netw.*, vol. 21, no. 4, pp. 654–666, 2008.

[19] R. Kobetic et al., "Development of hybrid orthosis for standing, walking, and stair climbing after spinal cord injury," *J. Rehabil. Res. Dev.*, vol. 46, no. 3, pp. 447–62, 2009.

[20] F. Zhang, M. Liu, and H. Huang, “Effects of locomotion mode recognition errors on volitional control of powered above-knee prostheses,” *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 23, no. 1, pp. 64–72, Jun. 2014.

[21] F. Sup, H. A. Varol, and M. Goldfarb, “Upslope walking with a powered knee and ankle prosthesis: Initial results with an amputee subject,” *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 19, no. 1, pp. 71–78, Oct. 2010.

[22] D. L. Grimes, “An active multi-mode above knee prosthesis controller,” Ph.D. dissertation, Dept. Mach. Eng., Massachusetts Inst. Technol., Cambridge, MA, USA, 1979.

[23] M. Goršič et al., “Online phase detection using wearable sensors for walking with a robotic prosthesis,” *Sensors*, vol. 14, no. 2, pp. 2776–2794, 2014.

[24] F. Zhang, Z. Fang, M. Liu, and H. Huang, “Preliminary design of a terrain recognition system,” in *Proc. Int. Conf. IEEE Eng. Med. Biology Soc.*, 2011, pp. 5452–5455.

[25] H. Kawamoto, S. Kanbe, and Y. Sankai, “Power assist method for HAL-3 estimating operator’s intention based on motion information,” in *Proc. 12th IEEE Int. Workshop Robot Human Interact. Commun.*, 2003, pp. 67–72.

[26] D. Novak et al., “Automated detection of gait initiation and termination using wearable sensors,” *Med. Eng. Phys.*, vol. 35, no. 12, pp. 1713–1720, 2013.

[27] E. Halilaj, A. Rajagopal, M. Fiterau, J. L. Hicks, T. J. Hastie, and S. L. Delp, “Machine learning in human movement biomechanics: Best practices, common pitfalls, and new opportunities,” *J. Biomech.*, vol. 81, pp. 1–11, Nov. 2018.

[28] S. Qiu, W. Guo, D. Caldwell, and F. Chen, “Exoskeleton online learning and estimation of human walking intention based on dynamical movement primitives,” *IEEE Trans. Cogn. Developmental Syst.*, vol. 13, no. 1, pp. 67–79, Mar. 2021.

[29] Y. Tu et al., “An adaptive sliding mode variable admittance control method for lower limb rehabilitation exoskeleton robot,” *Appl. Sci.*, vol. 10, no. 7, p. 2536, 2020.

[30] Z. Chen, Q. Guo, T. Li, Y. Yan, and D. Jiang, “Gait prediction and variable admittance control for lower limb exoskeleton with measurement delay and extended-state-observer,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 11, pp. 8693–8706, Nov. 2023.

[31] A. Taherifar, G. Vossoughi, and A. S. Ghafari, “Variable admittance control of the exoskeleton for gait rehabilitation based on a novel strength metric,” *Robotica*, vol. 36, no. 3, pp. 427–447, 2018.

[32] Y. Fei, J. Li, and Y. Li, “Selective memory recursive least squares: Recast forgetting into memory in RBF neural network-based real-time learning,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 36, no. 4, pp. 6767–6779, Apr. 2025.

[33] T.-H. Huang et al., “Modeling and stiffness-based continuous torque control of lightweight quasi-direct-drive knee exoskeletons for versatile walking assistance,” *IEEE Trans. Robot.*, vol. 38, no. 3, pp. 1442–1459, Jun. 2022.

[34] J. Park and I. W. Sandberg, “Universal approximation using radial-basis-function networks,” *Neural Comput.*, vol. 3, no. 2, pp. 246–257, 1991.

[35] P. Dhanalakshmi, S. Palanivel, and V. Ramalingam, “Classification of audio signals using SVM and RBFNN,” *Expert Syst. Appl.*, vol. 36, no. 3, pp. 6069–6075, 2009.

[36] Z. Chen et al., “RBFNN-based adaptive sliding mode control design for delayed nonlinear multilateral telerobotic system with cooperative manipulation,” *IEEE Trans. Ind. Informat.*, vol. 16, no. 2, pp. 1236–1247, Feb. 2020.

[37] P. Zhang, J. Zhang, and Z. Zhang, “Design of RBFNN-based adaptive sliding mode control strategy for active rehabilitation robot,” *IEEE Access*, vol. 8, pp. 155538–155547, 2020.

[38] N. T. Nguyen and N. T. Nguyen, *Model-Reference Adaptive Control*. Cham, Switzerland: Springer, 2018. [Online]. Available: https://link.springer.com/book/10.1007/978-3-319-56393-70

[39] J. B. Gomm and D. L. Yu, “Selecting radial basis function network centers with recursive orthogonal least squares training,” *IEEE Trans. Neural Netw.*, vol. 11, no. 2, pp. 306–314, Mar. 2000.

[40] H. P. Crowell, A. C. Boynton, and M. Mungiole, “Exoskeleton power and torque requirements based on human biomechanics,” Army Res. Lab., Adelphi, MD, USA, Rep. ARL-TR-2764, 2002.

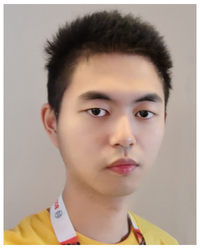
[41] D. Huang, J. Xia, C. Song, X. Xing, and Y. Li, “Path learning by demonstration for iterative human–robot interaction with uncertain time durations,” *IEEE Trans. Cogn. Develop. Syst.*, vol. 16, no. 2, pp. 436–447, Apr. 2024.

[42] X. Xing, S. Zhang, T. Huang, J. S. Huang, H. Su, and Y. Li, “Spatial iterative learning torque control of robotic exoskeletons for high accuracy and rapid convergence assistance,” *IEEE/ASME Trans. Mechatronics*, vol. 29, no. 6, pp. 4215–4227, Dec. 2024.

[43] D. A. Winter, *Biomechanics and Motor Control of Human Movement*. Hoboken, NJ, USA: Wiley, 2009.

[44] B. Chen, B. Zi, L. Qin, and Q. Pan, “State-of-the-art research in robotic hip exoskeletons: A general review,” *J. Orthopaedic Transl.*, vol. 20, pp. 4–13, Jan. 2020.

[45] K. Maqsood, J. Luo, C. Yang, Q. Ren, and Y. Li, “Iterative learning-based path control for robot-assisted upper-limb rehabilitation,” *Neural Comput. Appl.*, vol. 35, no. 32, pp. 23329–23341, 2023.

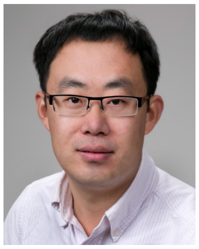


Wenlong Li received the B.Eng. degree in electronic and electrical engineering from Southwest Jiaotong University, Chengdu, China, in 2022. He is currently pursuing the Ph.D. degree in engineering with the Department of Engineering and Design, University of Sussex, Sussex, U.K.

His research interests include exoskeleton robot, human–robot interaction, and robot control.

Yiming Fei received the B.Eng. degree in control science and engineering and the M.S. degree in control science and engineering from the Harbin Institute of Technology, Harbin, China, in 2020 and 2023, respectively. He is currently pursuing the Ph.D. degree in artificial intelligence with Zhejiang University, Hangzhou, China.

His current research interests include reinforcement learning, cognitive neuroscience, neural network control, and robotics.



Hao Su (Senior Member, IEEE) received the B.S. degree in automation from the Harbin Institute of Technology, Harbin, China, in 2006, the M.S. degree in mechanical engineering from the State University of New York at Buffalo, Buffalo, NY, USA, in 2008, and the Ph.D. degree in mechanical engineering from Worcester Polytechnic Institute, Worcester, MA, USA, in 2013.

He is currently an Associate Professor with the Tandon School of Engineering, New York University (NYU), New York, NY, USA. Previously, he was an Associate Professor with the Department of Mechanical and Aerospace Engineering, North Carolina State University (NCSU), Raleigh, NC, USA, and held a joint appointment with the NCSU/UNC Joint Department of Biomedical Engineering, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA. He has also worked as a Research Scientist with Philips Research North America, Briarcliff Manor, NY, USA, and was a Postdoctoral Fellow with Harvard University, Cambridge, MA, USA, and the Wyss Institute for Biologically Inspired Engineering, Boston, MA, USA.

Dr. Su is a recipient of the National Science Foundation Career Award and the Switzer Distinguished Fellowship from the U.S. Department of Health and Human Services. He serves as a Technical Editor for the *IEEE/ASME Transactions on Mechatronics* and as an Associate Editor for the *IEEE Robotics and Automation Magazine*, the IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, and the *IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Qi Li received the B.S. degree in automation from Harbin Engineering University, Harbin, China, the M.S. degree in control science and engineering from Harbin Institute of Technology, China, and the Ph.D. degree in systems engineering from Nanjing University of Aeronautics and Astronautics, Nanjing, China.

He is the founder and CEO of VNCOL Technology Company Ltd., Xiamen, China.

Yanan Li (Senior Member, IEEE) received the B.Sc. degree in automation and the M.Sc. degree in control science and engineering from the Harbin Institute of Technology, Harbin, China, in 2006 and 2008, respectively, and the Ph.D. degree in robotics from the National University of Singapore, Singapore, in 2013.

From 2015 to 2017, he was a Research Associate with the Department of Bioengineering, Imperial College London, London, U.K. From 2013 to 2015, he was a Research Scientist with the Institute for Infocomm Research, Agency for Science, Technology and Research, Singapore. He is currently a Reader in robotics and the Director of the Centre for Robotics and Sensing Technologies (CROSS-Tech), University of Sussex, Brighton, U.K. His general research interests include human–robot interaction, robot control, and control theory and applications.