



Model-Learning-Based Partitioned Control of a Human-Powered Augmentation Lower Exoskeleton

Huu-Toan Tran¹ · Luy Nguyen Tan¹ · Seung-Hun Han²

Received: 22 April 2021 / Revised: 10 June 2021 / Accepted: 6 July 2021 / Published online: 25 August 2021
© The Korean Institute of Electrical Engineers 2021

Abstract

This paper presents a new model-learning-based partitioned control strategy of a wearable lower exoskeleton. Here, the dynamics of the coupled human-exoskeleton system along with the corresponding resulting interaction torques are learned based on nonparametric regression technique and then incorporated in the control system for swing phase. This promising combination of partitioned control scheme and incremental model learning has provided the exoskeleton with the ability to adapt to various dynamics of human operators, to reduce the physical interaction between the operator and the exoskeleton, and minimize the sensory system used in the system. In this method, movement data containing the information of dynamics and interaction was collected in a number of walking cycles, and then training and prediction procedure were performed to aid the controller. We have demonstrated the feasibility of the proposed method through an exoskeleton prototype that employs walking sessions on a bench-testing over different ranges of walking speeds (0.8–1.2 m/s) with various subjects. In the simulation results, the control performance of the proposed algorithm was qualitatively compared to other fundamental controllers including a classical impedance control and a Rigid-Body-Dynamics model-based control. The resulting interaction torque reduced to greater than 32%. These results were re-evaluated in the real system and similar performance was achieved.

Keywords Lower limb exoskeleton · Human–robot interaction · Wearable robotics · Computed torque control · Model learning · Non-parametric regression

1 Introduction

Lower exoskeletons are intelligent wearable robots which are worn by human operators as orthotic devices for performance assistance or work augmentation. They have attracted enormous attention in many fields such as labor-intensive industries, military and medicine. Although a number of studies relating to design of lower limb exoskeletons have been developed [1–8], many challenges continue to limit the performance of the system. One of the limiting factors is the lack of intelligent and effective control strategies for the incorporative human-exoskeleton system. In general, the aim

of the control strategies is to drive the lower exoskeleton to work in concert with the operator's movement while they fit closely to the body of the operators. To do this, the control of exoskeleton robots must satisfy the following requirements besides the normal expected performances in robotic field: (1) Capability of automatic gait-pattern adaptation since the physical characteristics of limb's segments change from person to person and also within one person over time; (2) Delivering assistive torques while reducing the human–robot interaction torques for natural movement of the operators as a key point compared to other traditional robots; (3) Rejecting disturbances due to un-modeled dynamics effects such as actuator dynamics, noise, and friction.

To deal with the above problems, many control methods have been developed, from conventional methods such as master–slave position control [1], state machine [9], virtual torque control [10], and impedance control [2] etc., to the enhanced methods such as admittance control [11], predictive control [12], primitive-based adaptive control [6], assist-as-need control [13], and EMG-based motion

✉ Seung-Hun Han
shhan@gnu.ac.kr

¹ Faculty of Electronic Technology, Industrial University of Ho Chi Minh City, Ho Chi Minh City 700000, Vietnam

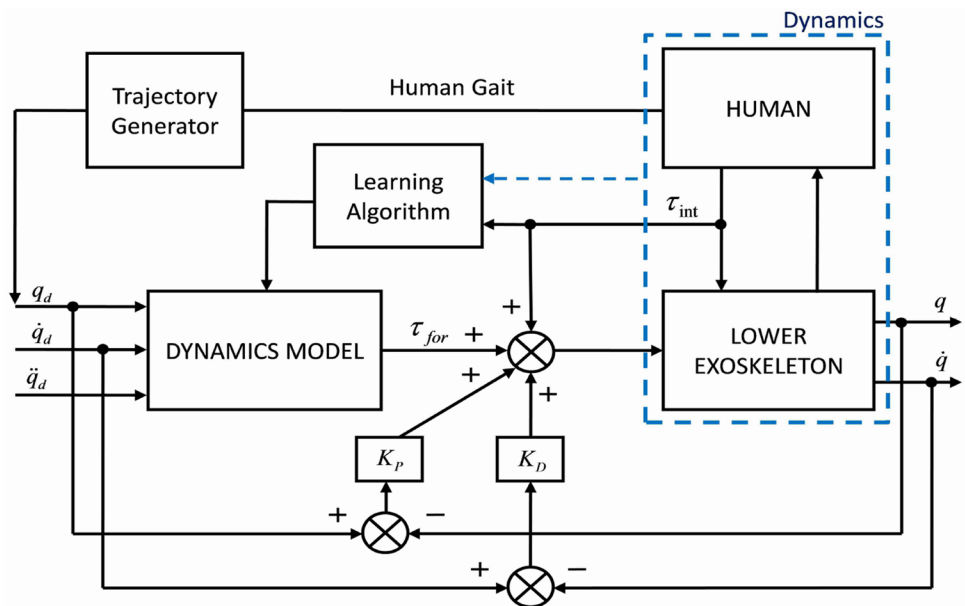
² Mechanical System Engineering Department, Tongyeong Campus, Gyeongsang National University, Tongyeong 53064, Republic of Korea

intention recognition control [14]. As one of the typical groups in improvement of control quality for the lower exoskeleton, Ugurlu et al. developed a newly active compliance control strategy where the force and position control loops are simultaneously managed to reduce the resulting interaction force and upper body effort [11]. Unlike traditional stiff control strategies as master-slave and impedance control, the proposed method computes the joint angles, contact force errors and force/position control law to provide compliant locomotion characteristics. The method depends significantly on hardware implementation and sensor calibration which are necessary to improve the clinical trials. In the scope of an exoskeleton at the knee joint, Staman et al. implemented a feedback force amplification on a novel series elastic element and electro-hydrostatic actuator to shadow the forces (torques) required from the human-operator muscle for control of the robot [15]. However, there are many simplifying assumptions for the calculation of feedback forces in this research that will be unreal when the robot cooperates with operators under the effects of uncertainties in external environments. One of the remarkable achievements of the Berkeley Lower Extremity Exoskeleton (BLEEX) system was to minimize the use of sensory information from human-exoskeleton interaction, and instead, utilized a novel control method, called Sensitivity Amplification Control (SAC) to add robustness to the system [1, 16]. However, to achieve the SAC control, the dynamic model of the BLEEX system has to be identified as well as possible with a large number of parameters. While, as a trade-off, the Hybrid Assist Limb (HAL) system used a number of sensing modalities, such as skin-surface Electro-Myo-Graphical (EMG), reaction and interaction force sensors, to input a specific control scheme, typically as EMG-based impedance control schemes [3]. Similarly, Gui et al. proposed an EMG-based assist-as-needed control method on their custom-built lower exoskeleton to estimate a subject's voluntary joint torque from a progressive learning strategy [13]. To perform these control schemes in the both projects, it was necessary to estimate operator's viscoelastic properties as the active interaction torques using the motion information measured from these sensors. Although the proposed control methods were generally not related to the complicated dynamic models, the design and calibration of the sensory system are not straightforward, especially in the case of the system attached to human during locomotion. By proposing a new concept of quasi-passive exoskeleton, Passive Exoskeleton Clothes of MIT exploited the quasi-passive elements of springs and variable damper to improve the efficiency of exoskeleton devices [9]. By similar approach, Yu et al. proposed an under-actuated wearable exoskeleton to carry a heavy load through a feasible modular-type wearable system and a sensing data estimation to synchronize the robot with a user [17]. A quasi-passive variable stiffness

spring exoskeleton system was also introduced to increased speed and reduce energy cost during walking [18]. In these investigations, only the exoskeletal structure is concentrated via damper and stiffness elements while a conventional state-machine controller was applied. Non-adaptive control and kinematic constraints in the musculoskeletal design hinder these robots from closely tracking human movement patterns. The proposal of newly emulated inertial compensation and active-impedance controls bring novel solutions to improve the agility of the exoskeletons in swing mode [19]. In a different scheme, a model of the full exoskeleton dynamics including the contacts with the environment have been withdrawn to compensate for the control strategy [20]. In general, the control performance in these works significantly depends on the design and calibration of the sensory system or on the computation cost of the exoskeleton dynamic model. Inspired by these points, we propose an intelligent control strategy of the coupled human-machine system with the ability to adapt to the change in the human-exoskeleton interaction, while reducing the number of sensors and substituting the issue of system identification.

Our goal in this paper is to develop a model-learning-based partitioned control strategy (MLPC) which utilizes a nonparametric regression technique (Meier et al. [21], Nguyen-Tuong et al. [22]) to learn the dynamics model of the human-exoskeleton system including the physical interaction torques for a partitioned control of the system as depicted in Fig. 1. By doing so, both dynamics nonlinearities and the resulting interaction torques would be compensated by the model being learnt. To collect data for learning, the operator-exoskeleton system is controlled by a simple position controller so that the exoskeleton can track the operator's movement in the first several gait cycles. Since control issues are frequently addressed in the swing leg which is subjected to large motions with high bandwidths contrary to the stance leg, the proposed strategy would be applied in the swing phase during walking while a position control with gravity compensation is used to control the stance exoskeleton leg. By using this approach of dynamic model learning, we can simultaneously deal with above obstacles in lower exoskeleton control. Firstly, nonparametric learning methods (Stulp et al. [23], Schaal et al. [24]) can approximate a mapping describing dynamics relationship that contains all nonlinearities of the combined human-exoskeleton system including interaction factors. Since there are no great differences of individual gait patterns (playing the role as excitation trajectory) from person to person as well as within person during walking, training and test trajectories for model learning would be guaranteed [25, 26]. While, from the viewpoint of system identification, the estimation of these dynamics parameters is not always straightforward due to the requirement of persistent excitation and the necessity of sufficiently rich data sets [25, 27]. Secondly, the

Fig. 1 The principle of model-learning-based partitioned control strategy (MLPC) for lower exoskeletons in swing phase using interaction-dynamics learning



interaction torques between the operator and the exoskeleton can be regarded as a part of nonlinear dynamics of the whole system. Not to count the force command from the human central nervous system, the force imposed by the human on the machine physically depends on the deviation between the joint position of the operator and exoskeleton which can be used as a learning input for MLPC [10, 28]. Finally, the sensors that are used to collect data and take as feedback signals of the entire system, are totally encoders on exoskeleton and inclinometers on human. This is because the proposed learning algorithm will only take inputs from joint angles, velocities and accelerations without any direct measurement of the interaction torques. Consequently, the aim of reducing the sensory system can be obtained.

The proposed control strategy would be evaluated on a HUman-powered Augmentation Lower EXoskeleton (HUALEX) developed at our center. HUALEX fundamentally aims to provide sagittal plane augmented torques at hip/knee joints and to support load-carrying (Fig. 4). We applied MLPC to perform walking sessions for different subjects (named A-C) wearing HUALEX on a bench-testing. The experimental results indicated that MLPC significantly improved the control performance with reduction in tracking error and the interaction torques, adaptation to novel dynamics and physical human properties yet within a predefined range of walking speeds. The remainder of this paper is organized as follows; firstly we generally present how the learning procedure of dynamic model including physical interaction factor can be incorporated into a partitioned control scheme of the exoskeleton. Subsequently, we present the implementation of MLPC on HUALEX. Finally, we test and evaluate the strategy in simulations as well as in

experiments. The results will point out the effectiveness of MLPC in the exoskeleton control.

2 Model-Learning-Based Partitioned Control (MLPC) Strategy

From the viewpoint of control strategy, the exoskeleton robots fall into either of two broad categories. The first is wearable robots augmenting the muscular force of healthy subjects and supporting load, as passive devices. Whenever the subject moves or works, the exoskeleton has to be commanded to follow his/her motion without any inhibition. The other is assistive robots that support patients with walking difficulties. For this type, the robot is an active device driven to follow predefined gait trajectories and combined with a bit of information from the patients. For a unified perspective on this study, we have focused mainly on the former, a lower exoskeleton for performance augmentation. On the other hand, from biomechanical studies of human behavior, the walking gait cycle is frequently divided into a number of distinct phases [29, 30]. Among them, for a single leg, it is divided into two phases: a load support stance phase and an unloaded swing phase. Here, the swing leg is subjected to large motions with high bandwidths contrary to the stance leg. The stance leg is also affected by ground reaction forces which obstruct the control implementation and evaluation. Therefore, to simplify learning-control scheme, the dynamic behavior of the leg while in swinging mode will be investigated, in which two-linkage revolute mechanism regarded as a 2-DoFs multi-link pendulum

(HUALEX leg in swing phase) will be used to evaluate the algorithm.

2.1 Control of Lower Exoskeletons with Model-Based Approach

Due to enabled high speed and compliant control, partitioned control (or computed torque control) were proposed and then developed adequately in the robotic field [31, 32]. However, the major obstacle of this control approach is the requirement of an accurate model for feed-forward torque generation. In the past, these models could be obtained using standard rigid body formulations, but this solution is limited due to model errors, actuator dynamics, and un-modeled nonlinearities [32, 33]. Moreover, it is not feasible to apply the solution for control of the combined human-exoskeleton system since the system is especially subjected to unpredictable human–robot interaction forces (torques). An alternative to analytically deriving the dynamic model as well as the interaction dynamics is to learn them. In swing phase, the human subject’s leg including HUALEX is generally modeled as a 2-DOF serial link mechanism (Fig. 2). The dynamics for this serial link are represented by the equation:

$$[M_E(q) + M_H(q)]\ddot{q} + [V_E(q, \dot{q}) + V_H(q, \dot{q})]\dot{q} + G_E(q) + G_H(q) + \tau_F + \tau_H = \tau_A, \tag{1}$$

where $q \in R^2$ is the generalized position vector comprising the hip and knee angles. The terms $M(q) \in R^{2 \times 2}$, $V(q) \in R^2$,

$G(q) \in R^2$ represent the positive definite inertia matrix, Coriolis and centrifugal torque vector, and gravity torque vector, respectively, of the human subject (index H) and the exoskeleton (index E). The term $\tau \in R^2$ of Eq. (1) is the torques acting onto the exoskeleton. Of them τ_A is the actuator torque vector of drives acting onto the links of the exoskeleton, τ_H is the torque vector originating from the operator-exoskeleton interaction, and τ_F is the friction torque vector around the joints of both the exoskeleton and operator. In general, τ_F are assumed as functions of joint positions and velocities as follows:

$$\tau_F = F(\dot{q}) = D_E\dot{q} + C_E\text{sign}(\dot{q}) + D_H(u(t))\dot{q}, \tag{2}$$

where D_E, C_E represent viscous friction and Coulomb friction coefficients around the joints of the exoskeleton respectively, D_H is the viscous friction coefficient around the joints of the operator, $u(t)$ is the muscle activation at the joints of the operator. In Eq. (2), the stiffness of the operator’s muscles is neglected since it is insignificant compared to other parameters.

The term τ_H are the torques generated by the muscles of the operator (resulting interaction torques) that physically result from deviations between the joint positions of the operator and exoskeleton. Given that, $\tilde{q} = q_h - q$ is the deviations of the desired joint angles of the operator from the actual ones of the exoskeleton during walking. The interaction torques from the operator τ_H can be described as the nonlinear functions of the deviation without damping. This assumption leads to the most vulnerable effect of the interaction torque that makes the system to be unstable. By doing so, the proposed control scheme will be designed under the least stable condition from the interaction:

$$\tau_H = H(\tilde{q}). \tag{3}$$

In general, the determination of structure and parameter of the nonlinear function $H(\tilde{q})$ depends on factors relating to the physical properties of a specific operator and how well the operator fits to the machine. Note that the sign of τ_H in Eq. (3) is the vector sign depending on direction and orientation of motion. The proposed algorithm will be validated with various cases of the H function when the exoskeleton is physically acted from various muscle acted from various models of each individual operator. For the sake of simplicity and convenience, Eq. (1) can be rewritten as:

$$M(q)\ddot{q} + V(q, \dot{q})\dot{q} + G(q) + F(\dot{q}) + H(\tilde{q}) = \tau_A, \tag{4}$$

where $M(q) = M_E(q) + M_H(q), V(q, \dot{q}) = V_E(q, \dot{q}) + V_H(q, \dot{q}), G(q) = G_E(q) + G_H(q)$.

Introducing vector K which is defined as: $K(q, \dot{q}, \tilde{q}) = V(q, \dot{q})\dot{q} + G(q) + F(\dot{q}) + H(\tilde{q})$.

Thus, the Eq. (4) becomes:

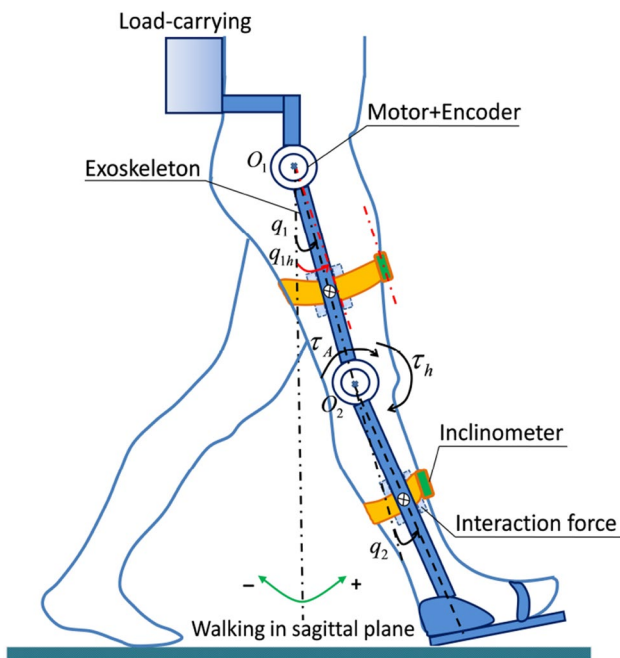


Fig. 2 An exoskeleton interacting with the operator’s lower limb during walking

$$M(q)\ddot{q} + K(q, \dot{q}, \ddot{q}) = \tau_A. \tag{5}$$

It can be seen from Eq. (4) that, if the dynamics of the system as well as the interaction factor $H(\ddot{q})$ can be estimated exactly, partitioned control method can eliminate the nonlinearities and the interaction dynamics as shown in Fig. 1. Using Locally Weighted Projection Regression, an inverse dynamic control can be achieved. This control is practically not exact because of the uncertainties and the resulting interactions in the combined human-exoskeleton system. The model-based learning control input can be written as [31, 32]:

$$\tau_A = \hat{M}(q)u + \hat{K}(q, \dot{q}, \ddot{q}) \tag{6}$$

where u is the new control input which makes the system to be robust and the notation $(\hat{\cdot})$ represents for the estimated corresponding item. The modelling error is expressed as:

$$\Delta M(q) = \hat{M}(q) - M(q), \quad \Delta K(q, \dot{q}, \ddot{q}) = \hat{K}(q, \dot{q}, \ddot{q}) - K(q, \dot{q}, \ddot{q}). \tag{7}$$

Now substituting the control (6) input into the human-exoskeleton model (5) yields:

$$\ddot{q} = u - \zeta(q, \dot{q}, \ddot{q}, u) \tag{8}$$

where the uncertainty is determined as:

$$\begin{aligned} \xi(q, \dot{q}, \ddot{q}, u) &= -M^{-1}(q) [\Delta M(q)u + \Delta K(q, \dot{q}, \ddot{q})] \\ &= [I - M^{-1}(q)\hat{M}(q)]u - M^{-1}(q)\Delta K(q, \dot{q}, \ddot{q}) \end{aligned} \tag{9}$$

The remaining task of the control scheme is to linearize the operator-exoskeleton system with a robust Proportional Derivative (PD) controller. For doing so, the control law is chosen as:

$$u = q_h + K_D(\dot{q}_h - \dot{q}) + K_P(q_h - q) + d \tag{10}$$

where $q_h, \dot{q}_h, \ddot{q}_h$ denote desired joint angles, velocities and accelerations of the human operator, K_p, K_v are two diagonal positive-define matrices, and d is an additional input to be determined. Here, the tracking error is defined as follows:

$$e = \begin{bmatrix} \tilde{q} \\ \dot{\tilde{q}} \end{bmatrix} = \begin{bmatrix} q_h - q \\ \dot{q}_h - \dot{q} \end{bmatrix} \tag{11}$$

By substituting the control laws (6) and (10) into the human-exoskeleton dynamic equation, closed-loop error equation is obtained as follows:

$$\dot{e} = Xe + Y(\xi - d) \tag{12}$$

where

$$X = \begin{bmatrix} 0 & I \\ -K_P & -K_D \end{bmatrix}, \quad Y = \begin{bmatrix} 0 \\ I \end{bmatrix}.$$

The robust PD control (10) leads to stabilize the nominal error system of a linear feedback, to compensate the accelerations of the lower limb joint angles, and to overcome the destabilizing effect of the uncertainty ξ . If K_P and K_D are two diagonal positive-definite matrices then X is Hurwitz. It is worth establishing a nominal error dynamics characterized by a frequency ω_i and a damping factor ζ . Therefore, the gains K_P and K_D can be chosen as follows:

$$K_P = \text{diag}(\omega_1^2, \omega_2^2), \quad K_D = \text{diag}(2\zeta_1\omega_1, 2\zeta_2\omega_2) \tag{13}$$

Clearly, as the configuration of the combined human-exoskeleton system changes, the effective close-loop gain changes and the poles move around in the real-imaginary plane. Nevertheless, we can use Eq. (13) to find a good set of constant gains such that, despite change of the poles, they are guaranteed to remain in favorable positions. This is equivalent to a starting point for the design of a robust controller [31, 32]. For example, using the Lyapunov’s methods, the control term d can be designed. In order to accomplish the robust model-based control for the combined human-exoskeleton system, a suitable expression for control law relating to $\|\xi\|$ has to be determined. In practice, $\|\xi\|$ is not adequate to determine because of the properties of the lower exoskeleton system in which there are uncertainties relating to the human biomechanics and human intention. Thus, this paper focused on the experimental evaluation to demonstrate the proposed control strategy. This approach is well known in the robotic lower exoskeleton field [1–9]. One of important remaining issues that will be discussed in next sections is to approximate the inverse dynamic model of the system for control law as Eq. (6). Note that, this control law is computed as a function of the desired trajectory only. Hence if the desired trajectory is known in advance, values usually can be computed off-line before the system begins working. However, in the case of controlling the exoskeleton to follow human motion, we need a trajectory generator which gets the kinematics information from human motion, such as human joint angles, velocities and accelerations. These signals will be directly computed based on the inclinometers attached on the operator limbs.

2.2 Key Features of Human–Robot Model Learning

If a reliable inverse dynamic model of the combined system is available and simultaneously the physical interaction between the operator and exoskeleton is predicted well, the proposed model-based control can produce much higher-quality control with less power consumption. This dynamic model is not easy to acquire even when using dynamic parameter estimation techniques due to persistent excitation issue [26]. Exciting trajectories in these techniques are unrealistic in normal operating regime of an exoskeleton. Therefore, a

more appropriate approach investigated in this paper is to use supervised learning techniques to approximate dynamic elements of the system. Looking at walking motion, the swing leg is usually subjected to large movement with high bandwidth while the interaction torques exerted by the human operator are correspondingly time varying. In order to approximate the dynamics of this high dimension system during quick and natural movement of the human operator, it needs incremental online training and adaptation learning. LWPR with the ability to incremental learning is particularly suited for this kind of regression problems since it was verified for real-time control applications [34–36].

2.2.1 Locally Weighted Projection Regression (LWPR)

In general, supervised learning techniques have been divided into two major categories: global and local methods. Global methods such as Neural Networks (Haykin et al. [37]), and Gaussian Process Regression (Schaal et al. [24]) fit nonlinear functions globally, provide a powerful tool to accurately approximate models from data. Local methods, typically as Locally Weighted Projection Regression (Nguyen-Tuong et al. [22]) make use of spatially localized models to fit a nonlinear function. LWPR is an incremental version of the locally weighted regression (LWR) algorithm with automatic structure adaptation and lower computational cost [38]. Compared to the global regression techniques, LWPR is appropriate for learning the complicated dynamic model containing the human–robot interaction factors of the system due to its fast learning speed [39, 40]. An LWPR model consists of a set of local linear models that come paired with a kernel that defines the area of validity of the local model. For a given input x , a weighting $\omega_k(x)$ is determined by the kernel of the k th local model while the local linear model predicts an output $\psi_k(x)$. The combined prediction of LWPR is calculated by the weighted average of N individual predictions:

$$\hat{y}(x) = \frac{\sum_{k=1}^N \omega_k(x) \psi_k(x)}{\sum_{k=1}^N \omega_k(x)} \quad (14)$$

with $\psi_k(x) = \bar{x}_k^T \hat{\theta}_k$ and $[(x - c_k)^T, 1]^T$. Here $\hat{\theta}_k$ contains the regression parameters and c_k is the center of the k th linear model. The region in which a linear model is valid is called the receptive field (RF). It is usually characterized by a Gaussian kernel:

$$\omega_k(x) = \exp\left(-\frac{1}{2}(x - c_k)^T D_k (x - c_k)\right) \quad (15)$$

where D_k is a positive definite matrix called distance metric. The main goal of the learning process is to adjust D_k and $\hat{\theta}_k$ so that the error between the predicted values and the targets is minimal. For learning the linear models $\psi_k(x)$, the regression parameter $\hat{\theta}_k$ is calculated by an online formulation of

weighted partial least squares (PLS) regression (Vijayakumar et al. [39]), instead of recursive least square (RLS) presented in [40]. The advantage of PLS compared to recursive least square (RLS) is to reduce the computing cost of regression since the PLS algorithm is done only in a subspace determined by principal components of input, instead of doing over the whole input space as in case of RLS [41]. The regression parameter $\hat{\theta}_k$ can be calculated recursively as detailed in [43]. The distance matrix D_k determines the locality of each local model that can be learned individually by the stochastic gradient descent. Based on a given cost function J_k , distance metric D_k is updated as the following rule:

$$D_k = M_k^T M_k, \quad \text{with} \quad M_k^{n+1} = M_k^n - \alpha \frac{\partial J_k}{\partial M_k} \quad (16)$$

where M_k is an upper triangular matrix that ensures D_k to be positive. Minimizing the penalized weighted mean squared error expressed in J_k , the distance metric D_k can be obtained [42]. The number of receptive fields is updated automatically. If a training data point x does not activate any RF by more than a constant ω_{gen} a new RF is created centered at x . It shows that the number of RFs is directly proportional to the complexity of the input. Parameter ω_{gen} is a tunable one called meta-parameter. Besides, there are several other parameters related to the convergence and approximation error of the algorithm that need to be tuned manually during learning [39, 42, 43].

2.2.2 Movement Data Collection

The main issue involved in the data collection for learning is that learned model is unable to generalize well outside the regions that it has been learned. In the field of robot manipulator control using model learning, in order to learn inverse dynamics of a robot and use it to control, training data and test data have to guarantee the learning procedure will not fall into overestimated domain. Besides, the actual joint angles of a robot have to be chosen such that the relationship between the inputs and the target are sufficiently nonlinear [22, 27]. These joint angles play the role of an exciting trajectory during training and a desired trajectory during predicting. Fortunately, these exciting and desired trajectories are guaranteed for the motion of a lower exoskeleton that follows a normal walking gait dominated by an operator. Taking only human walking into account, although the human gait is different within one person over time, this difference is not so significant and still ensures that the desired trajectory for prediction will be inside the regions that it has been learned. It means that movement data collected by that way is able to capture the nonlinear dynamics of the exoskeleton and the corresponding human–robot

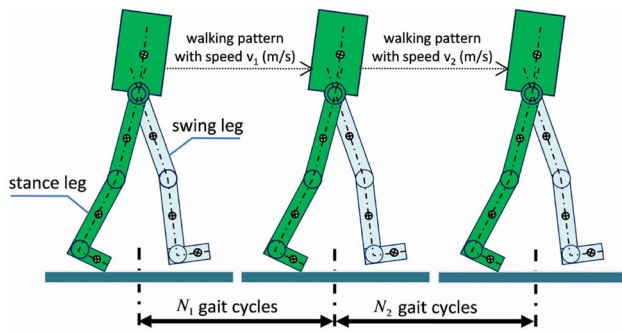


Fig. 3 Duration of data collection for learning and control: N_1 gait cycles for collecting data using a simple master–slave position control; N_2 gait cycles driven by MLPC

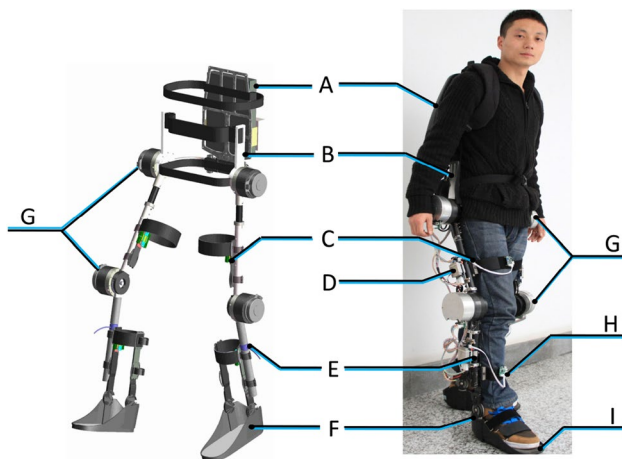


Fig. 4 Diagram of HUALEX's assembly and implementation. A: Backpack including load-carrying, power supply, and main controller; B: Rigid connection of the HUALEX to a user at torso; C: Two-dimensional Interaction Force Sensors (TIFs); D: Node controller and Elmo driver; E: Axial transfer force sensor (AFS); F: The HUALEX foot; G: Hip and knee actuators and encoders; H: Inclinator; I: Foot insole

interaction torques. These data, naturally, can be collected by controlling the exoskeleton to track the human motion with a non-model based controller, say a master–slave position controller.

As shown in Fig. 3, it is assumed that number ($N_1 + N_2$) iterations of the walking gait cycle (only considered in swing phase) will be performed. The exoskeleton is controlled by a simple master–slave PD controller with high gains in the first N_1 iterations of the trajectory and then switched to the partitioned controller with the model being learnt. Half of the observed data was used for training the model and half for testing and estimating the accuracy of the learned model. In order to obtain this adequate movement data, the design of the position master–slave controller for collecting data depends on several setup issues, for example, a proper

mechanical connection between the master (human) and the slave (exoskeleton) as exemplified in Fig. 2. Moreover, only data in swing phase collected for learning process requires a suitably sensory setup for phase detection as discussed in the next section (Fig. 4).

3 Implementation on Human-Powered Augmentation Lower Exoskeleton (HUALEX)

The control scheme discussed in Sect. 2 is motivated through simulations and experiments on HUALEX in this section. To deploy MLPC, the simplified configuration of HUALEX in swing phase is considered when the operator's leg with the exoskeleton is regarded as a two DOFs multi-link pendulum.

3.1 HUALEX Platform

From bio-mechanical studies of human behavior, HUALEX was designed as an anthropomorphic configuration [29, 44]. For simplification, there are totally ten-DOF (degree of freedom) revolute joints adopted for the HUALEX implementation (five DOFs for each leg). Among them, two-linkage revolute mechanism regarded as a 2-DOFs multi-link pendulum is actuated parallel to the human's thigh and shank for one leg. In order to implement and evaluate control strategies on HUALEX, thigh and shank cuffs mounted with the corresponding linkages through two-dimensional interaction force sensors (TIFs). The information from these sensors is not explicitly the entire force exerted on the HUALEX from human but is aimed at obtaining the change in the interaction force. TIFs can measure the deformation according to mainly flexion/extension orientations when external forces from a human are applied. For the sake of load distribution, a harness made of rigid structures is rigidly connected to a custom-built backpack to support the operator's upper body and transfer the carrying-load to the exoskeleton. The primary weight of the exoskeleton and carrying-load was aimed at 55 kg (40 kg for the load).

In order to allow the measurement of joint angles, encoders on HUALEX and inclinometers on human limbs were used. Incremental optical encoders (HP optical encoders HEDS) with a resolution of 2000 counts per revolution are adopted. For one leg, two custom-built inclinometers attached to the operator's shank and thigh enable to measure the angular position of his leg segments relative to the gravity. Each inclinometer was built using a low-cost MPU-6500 six-axis motion tracking sensor that combines a 3-axis gyroscope and a 3-axis accelerometer in a small-sized package and provides high resolution measurement (up to a scale range of $2000^\circ/s$ for gyroscope and scale range of $\pm 16 g$ for accelerometer). It is necessary to incorporate

the measurements of the acceleration in the long-term rate and the rotation in the short-term rate; hence, a low-cost micro control unit (MCU) (STM32F405, 32 bit, 168 MHz frequency) was used to control the inclination data stream and to implement Kalman filtering. Our application requires a fast communication and delay reduction in the impedance controller, so the inclination data was captured by MCU using SPI serial communications at 10 MHz. A least square method was used to calibrate the sensor. Figure 5a shows the performance verification displayed on a 3-D demo for one of the designed inclinometers.

To determine the phase of human gait, a set of Force Sensitive Resistors (FSRs) were distributed on the insoles. Two insoles were integrated into the exoskeleton shoes (Fig. 5b). The set of FSRs are located in a manner the static pressure distribution of the foot during walking is able to give as much information of the ground reaction force as possible [45, 46]. The function of the sensor-integrated insoles is absolutely important to distinguish the walking phases and to collect data for training and control. In MLPC, the switch of control modes between the stance phase and swing phase was immediately executed through the detected foot–ground information on the insole of HUALEX. Toe-off and heel-strike are the key indications of the switching between the stance and swing phases which were determined by groups of Force Sensitive Resistors (FSRs) at the toe and the heel of the insoles, as discussed in [47]. Besides, two axial force sensors (AFSS) was inserted into the exoskeleton shanks to measure the forces transferred from load-carrying through the exoskeleton which are able to validate the efficiency of the control strategy, as shown in Fig. 4. All the force sensors (FSRs, TIFSs, and AFSSs) were herein calibrated and tested using a standard analog force gauge (YueQing Handpi Instruments Co., LTD.). Besides, custom-built servo amplifiers are additionally equipped to measure the electrical power consumption for the actuators.

During development of the HUALEX, a new physically hierarchical control network (HCN) has been developed to increase the ability of integration, and ensure real-time

control performance. The architecture of HCN is a 3-level control network as shown in Fig. 6a consists of a central controller (master control) as the highest level, a node control level, and an I/O module. The master control communicates with the node control via Controller Area Network (CAN) to provide an understandable and consistent behavior and comfortable data visualization. In the central control level, a main STM32F ARM microcontroller programmed in C using ARM C/C++ Compiler has been developed. A graphical user interface developed on a personal computer (PC) using Visual C++ allowed us to observe experimental sessions and to acquire data for an offline learning processes, as seen in Fig. 6b. During the human–robot cooperation, safety feature for the system was also considered in the design of both the mechanical structure and software interruption mechanisms [47].

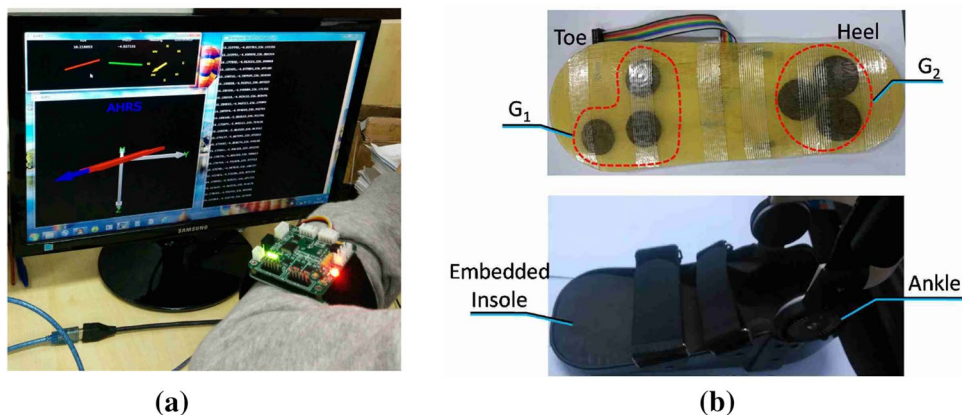
3.2 Analysis in Swing Phase

In order to apply the model learning approach to control of HUALEX, the inverse dynamics model that maps the knee angle, velocity, acceleration and the interaction torque to the actuator torque would be extracted. Equation (4) can be expressed as the form of a rigid body dynamics consisting of dynamic factor and interaction factor:

$$\tau_A = \underbrace{M(q)\ddot{q} + K(q, \dot{q})}_{\text{dynamic factor}} + \underbrace{H(\tilde{q})}_{\text{in interaction factor}} \quad (17)$$

Since the parameters M_H , V_H , G_H in Eq. (1) change according to varying operators, the matrix $M(q)$ and vector $K(q, \dot{q})$ will correspondingly change. Learning procedure will provide the system the adaptation to these changes. Besides, the interaction term $\tau_H = H(\tilde{q})$ that also changes from person to person needs to take information as a suitable input for the LWPR model. As discussed in Sect. 2, the equivalent interaction torque $H(\tilde{q})$ is generally represented as a nonlinear mapping that is mostly modeled as a spring type virtual coupling and is proportional to the deviation ($q_h - q$)

Fig. 5 Inclinometer and sensor-integrated foot: **a** Custom inclinometer and its performance verification on an animation demo; **b** The HUALEX insole (top) and foot (bottom); Force Sensitive Resistors (FSRs) are distributed on prototype HUALEX insole including two main groups G_1 , G_2 of ground reaction forces on the toe and heel



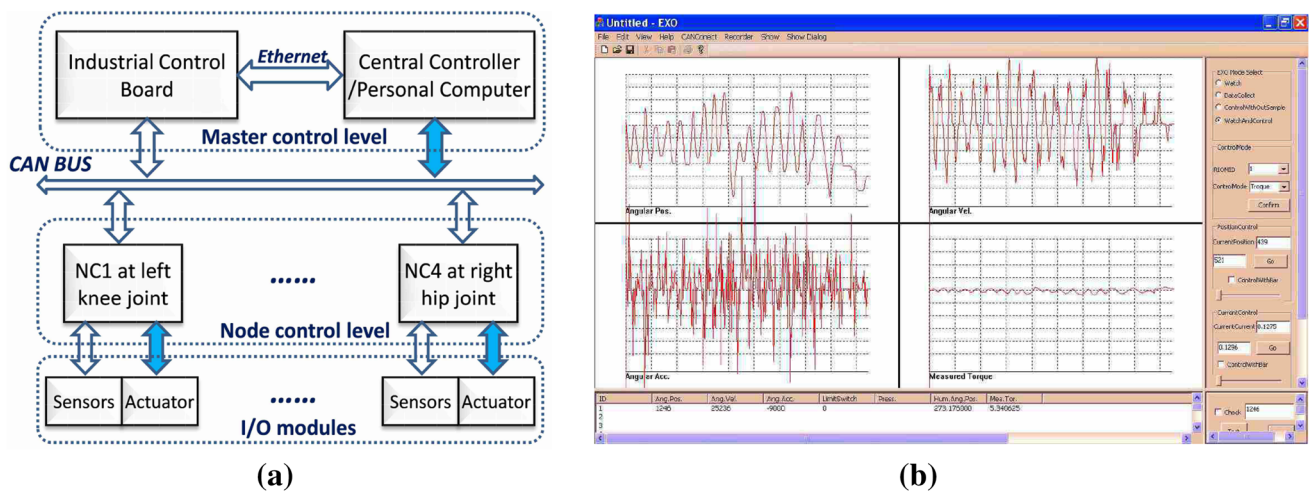


Fig. 6 Hardware configuration and user interface; **a** The designed hierarchical control network; **b** Designed interface for data visualization and collection

in the literature [2, 48]. However, this assumption might not be general. This is because, in reality, τ_H is an unpredictable function that depends not only on the deviation of the human–robot position but also on the physical properties of each wearer and the timing of joint movement. Here, the deviation is just taken as an input for the interaction factor learning. This choice is confirmed as a typical example [48]. The proposed control law would use this learned model to compensate for the torques exerted by the operator on the robot, without any measurement of this interaction torque.

3.3 Control Law Based on Learned Model

In order to approximate the mapping represented in Eq. (17), the inputs of LWPR network are the joint angles q , velocities \dot{q} , accelerations \ddot{q} of the exoskeleton and the deviation \tilde{q} between the joint angles of the operator and the exoskeleton, and the outputs are the actuator torques:

$$\tau_A = \hat{g}_{LWPR}(q, \dot{q}, \ddot{q}, \tilde{q}) \tag{18}$$

For a swing leg of the coupled human-exoskeleton system, the learning data can be obtained with a revolution optical encoder, an acceleration sensor on the exoskeleton and an inclinometer on the lower thigh of the human. The information from these measurements was continually stored and updated for training data batch. The operators walking pattern in the first cycles generates individual excitation trajectory for the training procedure. Subsequently, the corresponding learned model is aided the partitioned controller as follows:

$$\tau_A = \hat{g}_{LWPR}(q_h, \dot{q}_h, \ddot{q}_h, \tilde{q}^*) + K_p(q_h - q) + K_v(\dot{q}_h - \dot{q}) \tag{19}$$

The feed-forward command \hat{g}_{LWPR} given by the learned inverse model inputs the actual joint angles (q_h), the corresponding velocities (\dot{q}_h), accelerations (\ddot{q}_h) of the operator, and the current angle derivations (\tilde{q}^*) between the operator and exoskeleton joint angles. Whereas the feedback command, as discussed in Sect. 2, is provided by a PD controller to ensure that the system is stable with appropriate coefficients K_p , K_D . Through this equation, it can be seen that the more accurately the combined human–robot dynamic model can be learned, the higher the quality of compliant tracking control will be.

Regarding the master–slave position controller used here, it is not sufficiently appropriate during the swing phase since the exoskeleton is subjected to large motions and needs high bandwidth. In this case, it is easy to result in chattering phenomenon as well as large error in tracking control. Nevertheless, we accepted this choice as a trade-off for collecting data in the first several movements. For design of the master–slave position control, one of the key points is the consideration of the mechanical connection between the master (operator) and the slave (exoskeleton). Due to the fact that the controller takes the error between the master and the slave position as input, if we bind them together with rigid connections, the output signal for control goes to zero. That means it would impede the motion of the master. Therefore, the initial value of the inclinometers on the human limbs (shank and thigh) was set to 1.5 degree in order to avoid that case while the connections at thigh and shank cuffs were not rigid and only aimed to measure the change of the interaction forces. With this connection, the master–slave control of the exoskeleton is obviously guaranteed, but without good performance.

4 Evaluation of MLPC in Computer Simulations

4.1 Simulation Implementation

Since the gait patterns imposed by an operator are almost consistent during normal walking and can be customized to each individual, the collected data correspondingly contains his/her resulting interaction with the exoskeleton. To accomplish this, a suitable parameterization of the trajectories is used here to generate varying human gait trajectories [2, 49]. For simulation purposes, we choose these trajectories with different bound values and frequencies for training and prediction as follows:

$$q_d = Mq_n \left(\frac{t}{N} \right) + \varphi, \quad (20)$$

where M is the scale of the amplitude, N is the time stretch, φ is the offset of the individual gait, and q_n is the nominal physiological trajectories at hip and knee joints assumed to be superposition of sinusoidal trajectories. The goal of this choice is to validate the adaptation to various human gait patterns. The parameters M , N , φ would be changed according to the physical properties of each subject at each joint, as well as changed within one subject overtime during training and predicting. To simulate the generalization ability of MLPC, a 10% percent change in these parameters was introduced for each joint in swing phase. Regarding the structure and coefficients of the function describing interaction torque in Eq. (3), they were only estimated roughly such that they are close to the experimental results.

Three selected subjects (operators) in our center would take part in walking sessions for practical evaluation of the control strategy. Simulations were independently performed with these subjects. They are healthy persons whose weight were 73 kg (subject A), 65.5 kg (subject B), and 69 kg (subject C). The inertial and gravitational components of the lower limb (thigh and shank) of each subject (i.e. I_{Hi} , m_{Hi} , l_{Hi} , $i = 1$ for the hip joint and $i = 2$ for the knee joint) were calculated by means of experimental measurement and anthropometric data [29, 30]. We estimated these components which were based on percentage distribution of total body weight according to different segmentation plans. Besides, the dynamic parameters (e.g. I_{Ei} , m_{Ei} , l_{Ei} , $i = 1$ for the hip joint and $i = 2$ for the knee joint) were calculated by Solidworks from the design models of the HUALEX parts. The viscous friction and Coulomb friction coefficients of the exoskeleton were properly estimated from material and structure assembly of the experimental platform. The viscous friction D_H (affected by the activation muscle $u(t)$) was also artificially calculated from the define pattern [50]. Table 1 shows ranges of variation for input/output variable normalization of LWPR network in simulation. For

simulation, only 2-DOF human-exoskeleton model in swing phase was built. MLPC was evaluated using this model developed in Matlab (MathWorks, USA). The open-source LWPR was modified to incorporate the human-exoskeleton model for executing learning and control procedure [43].

For LWPR, its ability of learning incrementally robust and providing error bounds has been demonstrated [39, 43]. However, LWPR is sensitive to initial conditions and its convergence speed depends greatly on manual tuning parameters. In implementation, there are two parameters that sufficiently affect our learning results and need to be carefully selected. The first one is the distance metric matrix D whose initial value is associated with how large a receptive field is. The diagonal matrix D is initialized such that LWPR converges fast (value of D is not too small) as well as avoid over-fitting phenomenon of the learning data (value of D is not too large). In this case, it is set in the following form:

$$D = \frac{1}{2} \begin{pmatrix} 1/\sigma_1^2 & 0 & 0 & 0 \\ 0 & 1/\sigma_2^2 & 0 & 0 \\ 0 & 0 & 1/\sigma_3^2 & 0 \\ 0 & 0 & 0 & 1/\sigma_4^2 \end{pmatrix}, \quad (21)$$

where σ_i^2 is the variance of the i^{th} input (i goes from 1 to 4). Here, it is assumed that some prior information such as the range of operating angles or bound values of the deviations Δ is known for providing the distribution of the training input data. From this point, the matrix D is initialized using the inverse of the variance of these data. This is meaningful for our case when the training data have different distribution corresponding to various subjects (A to C). Besides, another parameter necessary for tuning is the weight activation threshold ω_{gen} , which is critical to the updated number of receptive field during learning. This meta-parameter is responsible for generating a new local model if no model responds high enough for a training sample. On the other hand, the LWPR inputs (q , \dot{q} , \ddot{q} , \tilde{q}) and output (τ_A) have to be normalized corresponding to their physical ranges. These ranges were selected based on the system performance when

Table 1 Ranges of variation for input/output variables normalization of LWPR network

Variable	Ranges of variation q_1	Ranges of variation q_2
Joint angle q (rad)	$[-\pi/4, \pi/4]$	$[0, \pi/2]$
Joint velocity \dot{q} (rad/s)	$[-2.5, 2.5]$	$[-3.5, 3.5]$
Joint acceleration \ddot{q} (rad/s ²)	$[-30, 30]$	$[-40, 40]$
Deviation $ q_h - q $ (rad)	$[0, \pi/18]$	$[0, \pi/18]$
Torque τ (Nm)	$[-20, 20]$	$[-15, 15]$

tracking the predefined motions. Table 1 shows our selection for these ranges.

4.2 Results and Analysis

We first demonstrate that an inverse dynamic model of the system along with resulting interaction factor can be learned online and then used for control. For each experiment regarding a specified subject (A, B, or C), twenty iterations of a number of walking cycles were carried out for the swing leg. The number of walking cycles is set to 5 by trial and error. In these first 5 motion cycles, the exoskeleton was controlled by the master–slave PD controller and the data was collected at the same time. The walking cycle is used over a typical period range of about 1.33 s to 2.0 s to ensure that system performance could be satisfied with various periods of the lower limb motion. Formula (20) is used for trajectory generation of both hip and knee joints but with different scale of the amplitude, time stretch, and nominal trajectories. The motion amplitudes were also constrained to a range of $\pm\pi/4$ rad for the hip joint and $[0 \quad \pi/2]$ rad for the knee. Half of the collected sample contained the data of the set $[q, \dot{q}, \ddot{q}, \Delta q]$ that is used for training the model and half for testing. For each subject, ten trials were performed. Results, Fig. 7a shows that the average of the normalized mean square error (nMSE) on the test data drops quickly through the 20 iterations and converges to a relatively small nMSE range (about 10^{-2}) for each subject. Where, the nMSE is defined as the mean square error divided by the variance of the target data values at each joint.

From Eq. (19), it can be seen that the accuracy of the learned model including the interaction factor can also be manifested by the ratio of feedback command (i.e. PD control command) to partitioned command (i.e. PD plus feed-forward control command). The smaller ratio could be

obtained, the more the nonlinearities in the system would be cancelled. The average of this ratio through 20 iterations shown in Fig. 7b indicates that LWPR can learn the system model with high accuracy already from the third iteration. Tracking performance in Fig. 8a shows that hip joint tracking error, in the presence of the interaction torque from the subject, is greatly reduced after fifteen walking cycles of learning (i.e. after 3 iterations). This result is achieved when we introduced 10% error in the motion amplitude and in the motion frequency. In this case, the PD coefficients of the initial PD control were not chosen with the best ones. As seen in Fig. 8b, the predictions of learned model as the feed-forward command at the hip joint are relatively close to the partitioned command. This demonstrates that a dynamic model including interaction factor was obtained by the proposed learning procedure. Furthermore, the effect of taking inputs for the learning was also validated. Control performance result with learned model using only normal dynamic information (i.e. q, \dot{q}, \ddot{q}) was compared with one using the aided information of interaction factor (i.e. $q, \dot{q}, \ddot{q}, \Delta q$). As seen in Table 2 (in the two last rows), if we only take the data set in the former case as inputs of LWPR, the learned model will be significantly less accurate than one in the latter case (Fig. 9).

The effect of learning the interaction factor in the partitioned control law was also evaluated through the performance comparisons to other methods under the same conditions of the motion pattern and interaction torque. Figure 10 shows the desired hip joint angle of subject B and the actual one of the exoskeleton after 50 swinging cycles (10 iterations) with a classical impedance control, a RBD model-based (without interaction compensator), and MLPC respectively. Here, impedance control which needs measurements of interaction forces (torques) is chosen to validate the efficiency of MLPC in reducing sensors. With respect to RBD model-based controller, a 15% error in

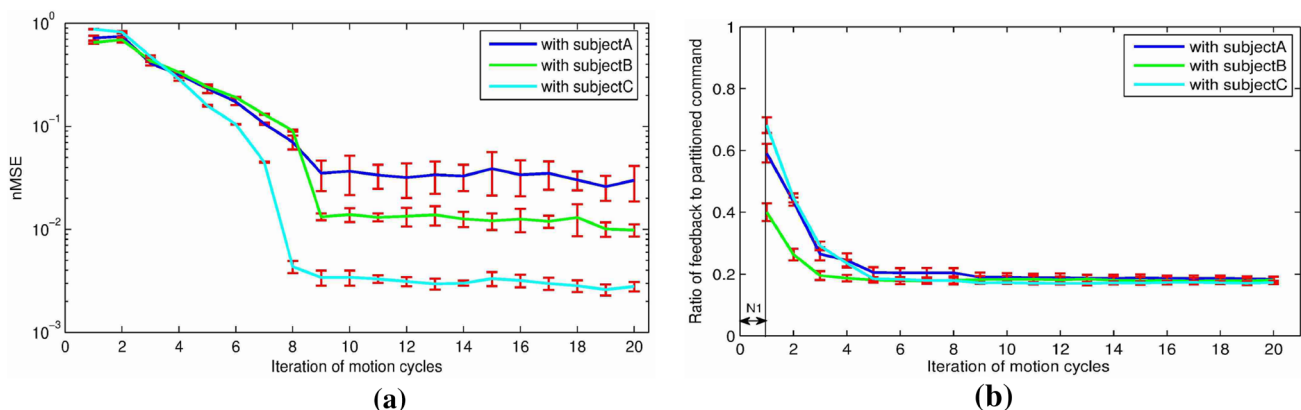


Fig. 7 Control performance during online learning using LWPR through motion cycles at knee joint. **a** nMSE on the test data for each subject; and **b** the PD control/partitioned control rate

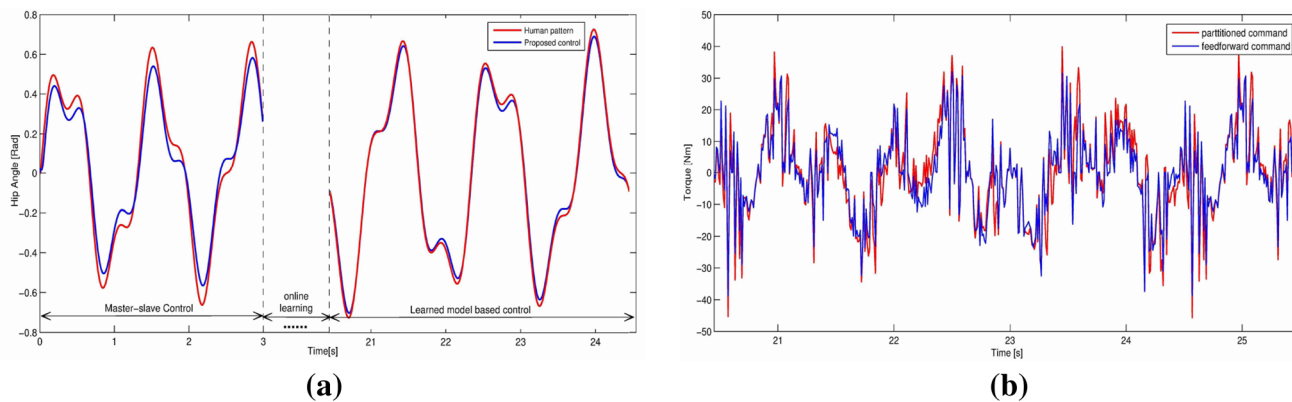


Fig. 8 MLPC tracking performance and partitioned command at the hip joint for subject B. **a** Tracking performance before and after fifteen motion cycles (3 iterations); **b** Comparison of feed-forward and

partitioned commands based on the model being learned as from the 15th motion cycle (20.5 s); Here, walking speed during training is 0.65 m/s and during prediction is 0.75 m/s

Table 2 Human-exoskeleton tracking error as nMSE

Tracking error (nMSE [%])	Subject A		Subject B		Subject C	
	Hip joint	Knee joint	Hip joint	Knee joint	Hip joint	Knee joint
Impedance control	5.39	5.08	5.91	4.54	6.04	5.87
RBD-model control	5.70	5.26	6.33	4.92	6.91	5.32
MLPC without int. fact	3.21 ± 0.53	3.15 ± 0.41	3.05 ± 0.27	2.85 ± 0.34	3.97 ± 0.25	3.69 ± 0.38
MLPC with int. fact	1.36 ± 0.32	1.22 ± 0.18	1.18 ± 0.15	1.03 ± 0.21	1.45 ± 0.19	1.31 ± 0.13

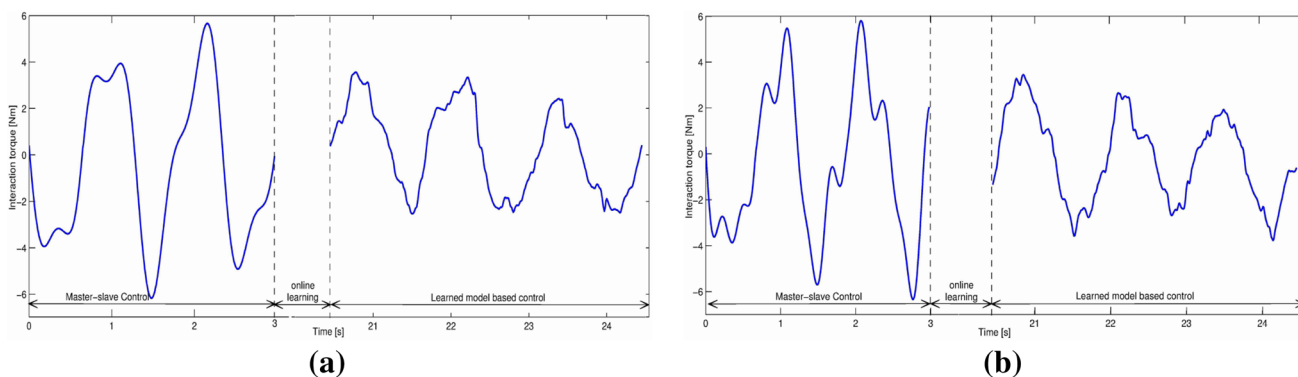


Fig. 9 Interaction torques before and after fifteen walking cycles of learning for subject B. Interaction torques were modeled by linear spring-like (a) and increasing nonlinear functions (b)

the inertial parameters was introduced. In Table 2, we can see how the proposed control law can partly compensate the dynamics and interaction effects while the impedance control cannot attenuate the interaction from human, and result in the larger tracking error at the hip joint. Similar results were also achieved at the knee joint. In Table 2, the terms “int. fact.” stands for “interaction factor”. It is more apparent to realize this effect in Fig. 8. The algorithm provided the robot with the ability to typically reduce the interaction torque greater than 30% whether this torque is described as a spring-like linear or an increasing nonlinear

function of the deviation \tilde{q} . This effect can be clearly seen after 3 iterations as from the position control switched to the partitioned control law (after 20.5 s) at the hip joint. Note that these results were achieved when movement pattern during training and predicting are not significantly different.

A drawback of simulation results in this case is that they cannot evaluate the function of load transfer through the stance leg which is one of the main targets in HUALEX. Besides, the assumption of the resulting interaction forces is impractical. Therefore, experimental sessions would be

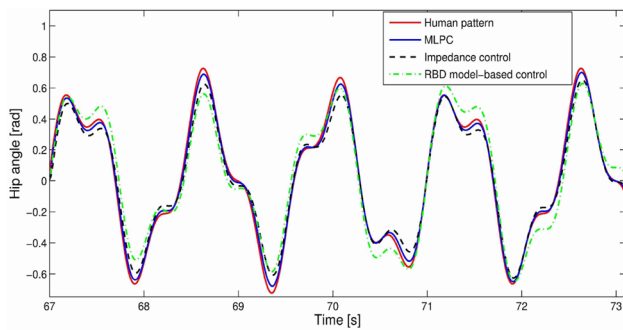


Fig. 10 Performance comparison of impedance control, RBD model based control, and the proposed method MLPC (after 50 walking cycles of learning)

progressed to validate MLPC on HUALEX, as discussed in the next section.

5 Experimental Evaluation

5.1 Experimental Procedure

In this work, a preliminary evaluation of the combined learning-control approach has been performed on real system as seen in Fig. 11. Only offline learned models have been used for tracking control of the human-exoskeleton system because online learning are incapable of executing on our embedded system. The online learning incorporated in control is significantly extensive to perform hence will be evaluated in the future work. In the experiments, the operators (A–C) were instructed by the author to wear HUALEX and to walk with a 40 kg load-carrying contained in the backpack at a given speed (0.6 m/s to 0.8 m/s) that was adjusted on the bench-testing monitor. Besides there are differences in anthropomorphic and dynamic parameters among the operators, their cadences which are determined to be from 75 to 80 steps/min at 0.75 m/s are also different from operators A to C. MLPC was applied for any leg in swing phase, therefore evaluation data was herein showed for the left leg as a representative. The operators alternately carried out the walking sessions and had break times among these sessions so that their muscles were not fatigued.

In order to collect data for training, duration of each session was conducted to be 15 s (8 to 12 strides) while duration of each session for testing the control methods is $T_c = 30$ s. As discussed above, each collected data set consists of four input vectors q , \dot{q} , \ddot{q} , \tilde{q} and one output vector τ_A . Note that the joint torques τ_A was obtained by measuring the motor currents using an inner PID current control of the driver. Besides, the interaction forces between the exoskeleton and operator at the thigh and shank were also stored for the subjective evaluation of experimental results. A common

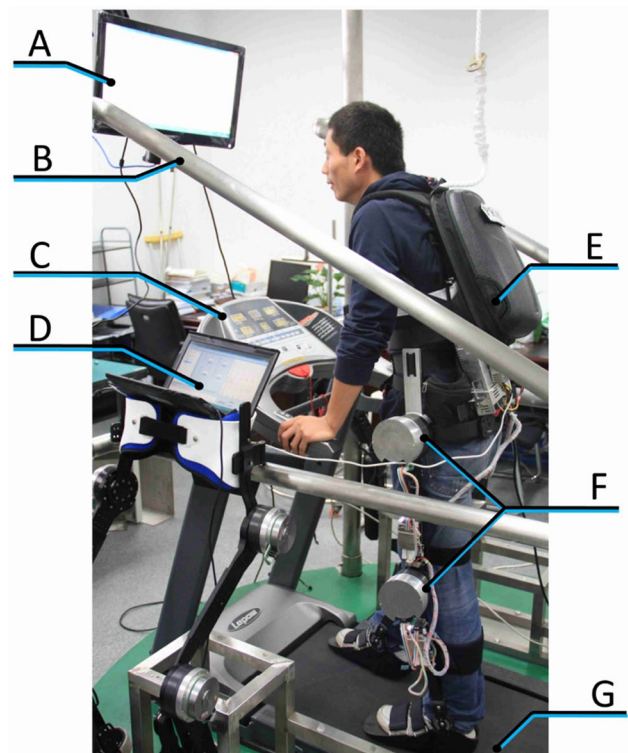


Fig. 11 Experimental session on the testing-bench. A: Monitor for visual feedback; B: Parallel bars; C: User Interface of testing-bench for modulating walking speeds; D: PC for data acquisition; E: Backpack; F: Hip and knee motors; G: testing-bench

impedance control was also implemented on HUALEX to validate the efficiency of MLPC in reducing sensors. This control approach come with the interaction force sensors at the shanks and thigh cuffs as feedback signals to regulate impedance characteristics according to the walking task, as discussed in [51].

The designed interface was used to display and store the collected data for each operator on the PC [45]. Due to the set of FSRs distributed on the exoskeleton insoles, only data captured from swing leg was stored for training. These sensors also served to distinguish walking phases and switch to a specific control mode (PD position control/ MLPC) corresponding to the current phase of the exoskeleton legs. The offline learning was also executed on the PC using the open-source LWPR package modified in C. After training, these learned models were then aided to control the exoskeleton under interacting with the corresponding subject in swing phase. Note that the position control was always utilized for stance leg. The feed-forward commands \hat{g}_{LWPR} were updated at every sampling step and the control algorithm was run at the sampling frequency of 100 Hz. This feed-forward command was computed based on joint position measurements (from the inclinometers, the encoders) and their differentiations. For each subject, five trials of training and control

were performed so that the control can be evaluated under the same conditions (before and after learning). For example, only the data gathered by the walking sessions with speed range of [0.6 m/s, 0.8 m/s] has been stored for training and then the subjects would be instructed to walking with the similar walking speeds constrained to the above range.

5.2 Results and Discussion

Figure 12 shows the human–robot tracking performances of the classical impedance control and MLPC lasting initial four seconds, while subject B performed walking session with specified speed of 0.75 m/s. Here, MLPC gets the off-line trained model from the data which was captured from his own walking sessions at speed of 0.9 m/s. It can be observed from the graph how the hip joint tracking error and the interaction force at the thigh cuff with MLPC decreases compared to the impedance control. The small difference in walking speeds for learning and control processes assure the learning would not be overestimated. In these experiments,

the axial transfer loads through stance legs were also stored as shown in Fig. 13. Both control methods satisfied the function of load transfer on HUALEX, yet it seems likely that MLPC brings more smoothly transfer through the exoskeleton than the impedance control. This explained why subject B could walk without any significant discomfort when wearing HUALEX controlled by MLPC.

In order to show the adaptation to different user dynamics, the responses of the exoskeleton using MLPC with different learned models were also compared. Figure 14 shows the human-exoskeleton tracking performance and the correspondingly resulting interaction forces in two experiments: (1) Subject B wearing HUALEX was instructed to walk at speed of 0.75 m/s in which MLPC is applied but with the learned model from experiments with subject A. (2) Similar to experiment 1 for subject B but MLPC is applied with the learned model from experiments with subject B. Tracking performance in Fig. 14b show that the tracking error and interaction force is significantly reduced when learned dynamics model is corresponding to the current subject.

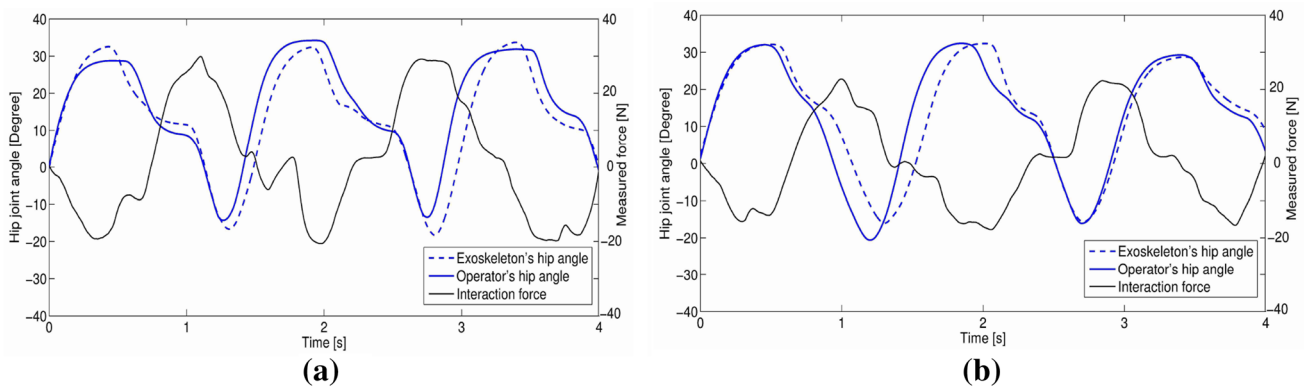


Fig. 12 Operator-exoskeleton tracking performance and the correspondingly resulting interaction forces when using different control methods. The performance was obtained from experiments with Sub.B around the hip joint at walking speed of 0.75 m/s using the impedance control (a) and MLPC (b)

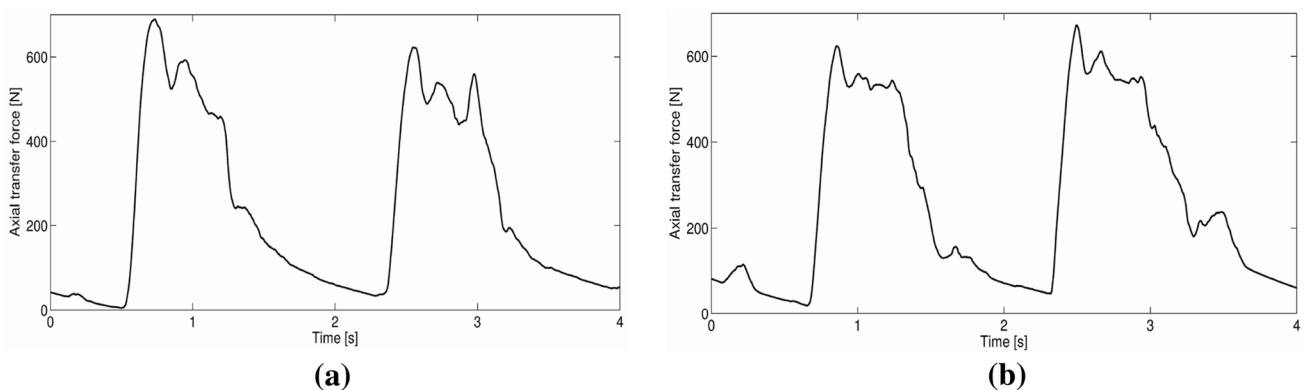


Fig. 13 Axial transfer force during walking when using the impedance control and MLPC. The data was obtained from experiments with Sub.B using the impedance control (a) and MLPC (b) corresponding to Fig. 12

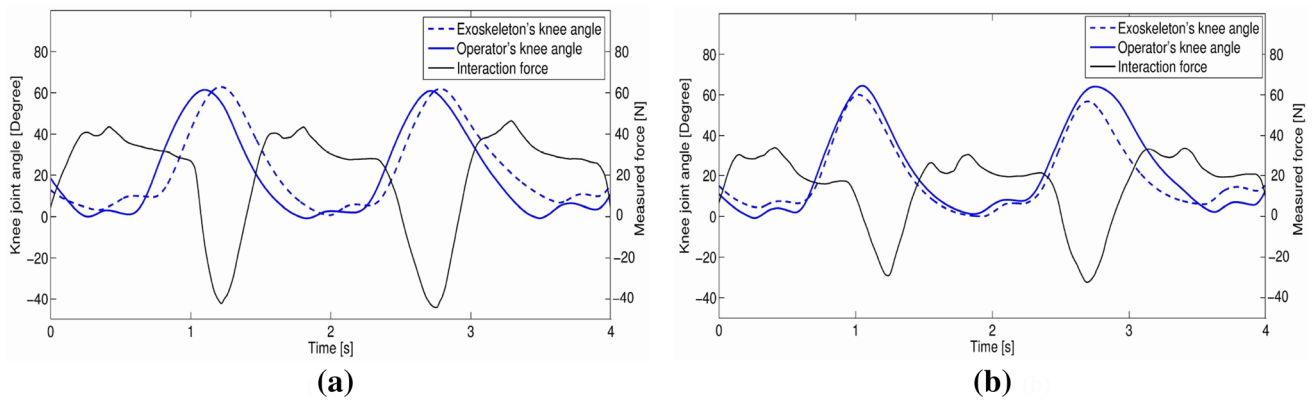


Fig. 14 Operator-exoskeleton tracking performance and the correspondingly resulting interaction forces when using MLPC with different learned models. The performance was obtained from experiments with Sub.B around the knee joint at walking speed of 0.65 m/s using (a) MLPC with incompatible model (from Sub. A); (b) MLPC with compatible model

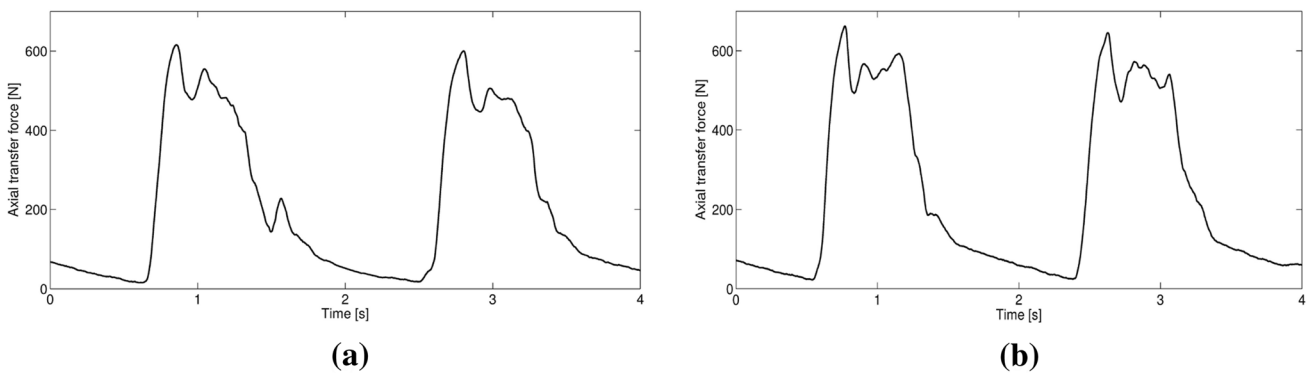


Fig. 15 Axial transfer force during walking when when using MLPC with different learned models corresponding to Fig. 14. (a) MLPC with incompatible model; (b) MLPC with compatible model

On the contrary, MLPC implemented with incompatible models, as seen in Fig. 14a, produced many discomforts due to large resulting interaction and low compliance. The term “incompatible models” is defined as the models are learned from different individuals, or the models were not captured enough information about dynamics and interaction properties. These control performances demonstrate that learned models indeed contained rich information about the individual dynamics of each own user, and that MLPC provided the exoskeleton with ability to adapt to changes in the corresponding dynamic information. Moreover, due to the above discomfort, MLPC with incompatible model interfered with the load-carrying transfer to the ground through HUALEX as seen in Fig. 15. The load transfer efficiency is reduced 15–20% when we used MLPC with incompatible model compared to that with compatible model.

To quantitative comparison, performance index of the resulting interaction forces for the control methods are shown in Table 3 in which it can be seen that the resulting interaction forces when using MLPC with compatible

model are significantly smaller than the impedance control and MLPC with incompatible model. In this table, performance index of the resulting interaction force is defined as follows:

$$F_i = \frac{\int_0^{T_c} f_{hi}^2(t)dt}{\int_0^{T_c} q_{hi}^2(t)dt}, \tag{22}$$

where $F_i(i = 1,2)$ is the normalized square sum of the sagittal plane interaction force at the thigh cuff (F_1) and the shank cuff (F_2). F_i are averaged over $T_c = 30$ seconds walking for subjects A-C. In Table 3, we can see that F_1, F_2 reaches the lowest value of around 3.21; 1.15 respectively with the case of MLPC using compatible model. In experiments, the RBD model based control related to some works on identification has not been implemented for the comparison since it is inconsistent with our approach. Although it can be seen that the interaction torque did not regularly changed in all walking cycles as shown in Fig. 12, it is typically reduced greater than 25% to 30% when using the proposed MLPC.

Table 3 Performance index of interaction forces

Performance index of interaction forces	Subject A		Subject B	
	Thigh	Shank	Thigh	Shank
Impedance control	4.19 ± 0.37	1.78 ± 0.29	3.91 ± 0.25	1.54 ± 0.19
MLPC with incompatible model	4.27 ± 0.62	1.92 ± 0.22	4.06 ± 0.67	1.76 ± 0.27
MLPC with compatible model	3.54 ± 0.24	1.26 ± 0.10	3.21 ± 0.18	1.15 ± 0.12

Clearly, as the dynamics and the interaction factors have been learned, the controller successfully acquires the ability to predict and compensate for these factors during operator's motion, and effective closed-loop control response was achieved.

6 Discussion and Conclusion

We have developed a control algorithm that takes advantage of the incremental learning technique to improve the control of the lower exoskeleton and to compensate the resulting physical human–machine interaction. Satisfactory results of the simulation and experiments on a prototype of developed lower exoskeleton have been achieved in almost all cases, with the convergence of the learning algorithm and the significant reduction of the interaction torque. Compared to other conventional control methods, the proposed method provided considerable compliance in human–robot tracking error, and thus the exoskeleton could assist/support the operator without discomfort. Nevertheless, the results from the experiments in all cases were not as good as those from the simulation. This is because the control performance during experimental sessions was affected by the error in calibration process of the inclinometer and accelerometer, and time delay to reach operator's pattern. Especially, only offline learned model of the human–robot system used for control of real system is not sufficiently efficient, since the dynamics and interaction factors continually change during locomotion. This encourages us to validate the online learning process on the real system with the need to improve hardware configuration in the future work.

During implementation of this investigation, we have faced three major drawbacks. The first one is how to achieve rich enough information in order to significantly capture the dynamic property of the whole system in a wide range of motion. Although we are assured that the test data is different from the training data, the limitation of this choice is that the difference is not so large. For example, when we chose the walking speed of 1.0 m/s (normal walking movement) for training data, the prediction from learned model

could not be over a range of [0.75 m/s, 1.25 m/s] (equivalent to running movement). It was less accurate when we performed the experiments with a large difference in these values before and after learning, even the prediction error did not converge. Besides, the initial duration N_1 (using the master–slave control) was appropriately chosen because it affects the computation cost and the motion periods in which the human–exoskeleton system is subject to certain difficulties and discomfort. The second challenge is the trade-off between the accuracy and computation cost of the nonlinear regression technique. The literature showed the comparisons of learning accuracy and computation speed to present us with a suitable choice among state-of-the-art nonparametric regression networks [22, 52]. As a preamble for combination of a control scheme for the exoskeleton and model learning, this study utilized a local regression technique that provides high computation speed but less accuracy compared to other global learning methods (e.g. GPR). Moreover, the manual tuning for parameters of the algorithm, such as metric D or coefficient ω_{gen} give rise to tediousness for designers by trial-and-error. This also encourages us innovate the above regression methods in order to enhance the efficiency of these methods in the application of the human–robot system. Finally, in order to evaluate the proposed algorithm, the tracking error, resulting interaction forces, and transfer forces have been provided as basic criterions. However, with respect to the human-enhancement system like the lower exoskeleton, it would be more persuasive if we expand the evaluation to flexor–extensor muscle of the operator, or metabolic effect of forces exerted onto the operator during movement.

References

1. Kazerooni H, Steger R, Huang L (2006) Hybrid control of the Berkeley lower extremity exoskeleton (BLEEX). *Int J Robot Res* 25(5–6):561–573
2. Riener R, Lünenburger L, Jezernik S et al (2005) Patient-cooperative strategies for robot-aided treadmill training: first experimental result. *IEEE Trans. Neural Syst Rehabil Eng* 13(3):380–394

3. S. Lee and Y. Sankai (2002) Power Assist Control for Walking Aid with HAL-3 Based on EMG and Impedance Adjustment around Knee Joint. In: Proceedings of IEEE International Conference on Intelligent Robots and Systems, Lausanne Switzerland, pp 1499–1504
4. Veneman JF, Kruidhof R, Hekman EEG et al (2007) Design and evaluation of the LOPES exoskeleton robot for interactive gait rehabilitation. *IEEE Trans. Neural Syst Rehabil Eng* 15(3):379–386
5. Li Y, Guan X, Han X et al (2020) Design and preliminary validation of a lower limb exoskeleton with compact and modular actuation. *IEEE Access* 8:66338–66352
6. Nunes PF, Ostan I, Siqueira AAG (2020) Evaluation of motor primitive-based adaptive control for lower limb exoskeletons. *Front. Robot. AI* 7:575217
7. Ruiz Garate V, Parri A, Yan T et al (2017) Experimental validation of motor primitive-based control for leg exoskeletons during continuous multi-locomotion tasks. *Front Neurobot* 11:15
8. Chao-Feng C, Zhi-Jiang D, Long H et al (2019) Development and hybrid control of an electrically actuated lower limb exoskeleton for motion assistance. *IEEE Access* 7:169107–169122
9. Walsh CJ, Endo K, Herr H (2007) A quasi-passive leg exoskeleton for load-carrying augmentation. *Int J Humanoid Rob* 4(3):487–506
10. Racine JL (2003) Control of a lower extremity exoskeleton for human performance amplification, doctoral diss., Berkeley, CA: University of California
11. Ugurlu B, Oshima H, Sariyildiz E (2020) Active compliance control reduces upper body effort in exoskeleton-supported walking. *Int J Biomed Biol Eng IEEE Trans Human Mach Syst* 50(2):144–153
12. Cao Y, Huang J (2020) Neural-network-based nonlinear model predictive tracking control of a pneumatic muscle actuator-driven exoskeleton. *IEEE/CAA J Automatica Sinica* 7(6):1478–1488
13. Gui K, Tan U-X, Liu H (2020) Electromyography-driven progressive assist-as-needed control for lower limb exoskeleton. *IEEE Trans Med Robots Bionics* 2(1):50–58
14. Villa-Parra AC, Delisle-Rodriguez D, Botelho T et al (2018) Control of a robotic knee exoskeleton for assistance and rehabilitation based on motion intention from sEMG. *Res Biomed Eng* 34(3):198–210
15. Staman K, Veale AJ, van der Kooij H (2021) Design control and evaluation of the electro-hydrostatic actuator, PREHydra for gait restoration exoskeleton technology. *IEEE Trans Med Robot Bionics* 3(1):156–165
16. Kazerooni H, Racine JL et al. (2005) On the control of the Berkeley lower extremity exoskeleton (BLEEX). In: Proceedings of IEEE international conference on robotics and automation, Barcelona, Spain, pp 4353–4360
17. Yu SN, Lee HD, Lee SH et al (2012) Design of an under-actuated exoskeleton system for walking assist while load carrying. *Adv Robot* 26(5–6):561–580
18. Sutrisno A, David J (2019) Braun, enhancing mobility with quasi-passive variable stiffness exoskeletons. *IEEE Trans Neural Syst Rehabil Eng* 27(3):487–496
19. Aguirre-Ollinger G, Colgate J, Peshkin M, Goswami A (2012) Inertia compensation control of a one-degree-of-freedom exoskeleton for lower-limb assistance: initial experiments. *IEEE Trans Neural Syst Rehabil Eng* 20(1):68–77
20. Vantilt J, Tanghel K, Afschrift M et al (2019) Model-based control for exoskeletons with series elastic actuators evaluated on sit-to-stand movements. *J NeuroEng Rehabil*, vol 16, no 65
21. Meier F, Hennig P, Schaal S (2014) Incremental local gaussian regression. In: Advances in Neural Information Processing Systems, pp 972–980
22. Nguyen-Tuong D, Peters J, Seeger M (2008) Computed torque control with nonparametric regression models. In: Proceedings of 2008 American Control Conference, Seattle, American, pp 212–217
23. Stulp F, Sigaud O (2015) Many regression algorithms, one unified model: a review. *Neural Netw* 69:60–79
24. Schaal S, Atkeson CG, Vijayakumar S (2002) Scalable techniques from nonparametric statistics for real-time robot learning. *Appl Intell* 17:49–60
25. Ghan J, Steger R, Kazerooni H (2006) Control and system identification for the berkeley lower extremity exoskeleton (BLEEX). *Adv Robot* 20(9):989–1014
26. Petkos G, Toussaint M, Vijayakumar S (2006) Learning multiple models of non-linear dynamics for control under varying contexts. In: Proceedings of 16th International Conference on Artificial Neural Networks, Greece, pp 898–907
27. Armstrong B (1989) On finding exciting trajectories for identification experiments involving systems with nonlinear dynamics. *Int J Robot Res* 8:28–48
28. Tran HT, Cheng H, Lin XC et al (2014) The relationship between physical human-exoskeleton interaction and dynamic factors: using a learning approach for control applications. *Sci China Inf Sci* 57(12):1–13
29. Winter DA (2009) Biomechanics and Motor Control of Human Movement, 4th edn. John Wiley and Sons Inc, New Jersey
30. Anthropometric source book, Volume I: Anthropometry for designers, (NASA Ref. Pub: Webb Associates, No. 1024, 1978)
31. Craig JJ (2004) Introduction to Robotics: Mechanics and Control. Prentice Hall, New Jersey
32. Kelly R, Santibáñez V, Loria A (2005) Control of robot manipulators in joint space. *Adv Textbooks in Cont and Sig Proc Series*, Springer, London
33. Khosla PK (1988) Some experimental results on model-based control schemes. In: Proceedings of IEEE International Conference on Robotics and Automation, America, pp 1380–1385
34. Nguyen-Tuong D, Peters J (2011) Model learning for robot control: a survey. *Cognitive Processing Springer* 12:319–340
35. Gao T, Yin S, Gao H et al (2018) A locally weighted project regression approach-aided nonlinear constrained tracking control. *IEEE Trans Neural Netw Learn Syst* 29(12):5870–5879
36. Ho JDL, Lee K-H, Tang WL et al (2018) Localized online learning-based control of a soft redundant manipulator under variable loading. *Adv Robot* 32(21):1168–1183
37. Haykin S (1999) Neural network: a comprehensive foundation, 2nd edn. Prentice Hall, New Jersey
38. Atkeson CG, Moore AW, Schaal S (1997) Locally weighted learning. *Artif Intell Rev Spec Issue Lazy Learn* 11:11–73
39. Vijayakumar S, Schaal S (2000) Locally weighted projection regression: an O(n) algorithm for incremental real time learning in high dimensional space. In: Proceedings of the 16th Conference on Machine Learning, San Francisco, CA, USA, pp 1076–1089
40. Nguyen-Tuong D, Seeger M, Peters J (2009) Model learning with local Gaussian process regression. *Adv Robot* 23:2015–2034
41. Nakanishi J, Farrell JA, Schaal S (2005) Composite adaptive control with locally weighted statistical learning. *Neural Netw* 18:71–90
42. Vijayakumar S, D'Souza A, Schaal S (2005) Incremental online learning in high dimensions. *Neural Comput* 12:2602–2634
43. Klanke S, Vijayakumar S, Schaal S (2008) A library for locally weighted projection regression. *J Mach Learn Res* 9:623–626
44. Rose J, Gamble JG (2006) Human walking, 3th edn. Williams & Wilkins, Baltimore
45. Bamberg SJM, Benbasat AY, Scarborough DM et al (2008) Gait analysis using a shoe-integrated wireless sensor system. *IEEE Trans Inf Technol Biomed* 12(4):413–423

46. Shu L, Hua T, Wang Y et al (2010) In-shoe plantar pressure measurement and analysis system based on fabric pressure sensing array. *IEEE Trans Inf Technol Biomed* 14(3):767–775
47. Tran HT, Cheng H, Huang R et al (2016) Evaluation of a fuzzy-based impedance control strategy on a powered lower exoskeleton. *Int J Soc Robot* 8:103–123
48. Jezernik S, Colombo G, Morari M (2004) Automatic gait-pattern adaptation algorithms for rehabilitation with a 4-DOF robotic orthosis. *IEEE Trans Rob Automation* 20:574–582
49. Kim JH, Han JW, Kim DY et al (2013) Design of a walking assistance lower limb exoskeleton for paraplegic patients and hardware validation using CoP. *Int J Adv Robotic Syst* 10:113–125
50. Erdemir A, McLeanet S, Herzog W et al (2007) Model-based estimation of muscle forces exerted during movements. *Clin Biomech* 22:131–154
51. Tran H-T, Cheng H, Duong M-K et al (2014) Fuzzy-based impedance regulation for control of the coupled human-exoskeleton system. In: *IEEE international conference on robotics and biomimetics (ROBIO 2014)*, pp 986–992
52. Nguyen-Tuong D, Peters DJ (2011) Incremental online sparsification for model learning in real-time robot control. *Neurocomputing* 74:1859–1867

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.