Fuzzy cerebellar model articulation controller-based adaptive tracking control for load-carrying exoskeleton

Han Wu¹, Lin Lang², Honglei An¹, Qing Wei¹ and Hongxu Ma¹

Abstract
Load-carrying exoskeletons need to cope with load variations, outside disturbances, and other uncertainties. This paper proposes an adaptive trajectory tracking control scheme for the load-carrying exoskeleton. The method is mainly composed of a computed torque controller and a fuzzy cerebellar model articulation controller. The fuzzy cerebellar model articulation controller is used to approximate model inaccuracies and load variations, and the computed torque controller deals with tracking errors. Simulations of an exoskeleton in squatting movements with model parameter changes and load variations are carried out, respectively. The results show a precise tracking response and high uncertainties toleration of the proposed method.

Keywords
Load-carrying exoskeleton, fuzzy cerebellar model articulation controller, adaptive tracking control, load variation

Introduction
The exoskeleton is a kind of wearable and anthropomorphic device that enhances the physical capabilities of a non-disabled wearer.¹,² It looks like a humanoid robot or a part of that in appearance and is worn in parallel with human body in general. The significant feature of the mechanical device is that it combines the cognitive ability of human with powerful mechanical system,³ which makes the human–machine system more intelligent and powerful. For example, lower limb exoskeleton can adapt complex terrains well without building map in advance.⁴ Load-carrying exoskeleton (also called load-bearing or load transfer exoskeleton) is one kind of lower limb exoskeletons. It transfers substantial loads to the ground directly rather than augments human joints power.¹ This type of leg exoskeleton is urgently needed in the military for long-distance locomotion with heavy loads⁵ and is widely researched during the past two decades. However, one of the most difficult problems in designing an excellent load-carrying exoskeleton is the control system. On the one hand, the device should follow the pilot’s motion flexibly. On the other hand, it should overcome load variation and other disturbances and support the remainder loads steady at the same time. In most cases, the carried loads could be a few times heavier than the device itself, and the variation of loads is a tough challenge for control system design.

Much work has been done on the control of load-carrying exoskeleton by earlier studies. But most of them deal with fixed payload or small load variation, or only focus on the uncertainty of the dynamic model. Among them, more effective and practical methods include sensitivity amplification control (SAC), model-based control (MBC), sliding mode control (SMC), and neural network-based adaptive control (NNAC). The Berkeley Lower Extremity Exoskeleton (BLEEX) is the first functional load-carrying activated exoskeleton. It uses SAC method to acquire a high sensitivity of pilot’s forces or torques without any human side information, which is hard to be measured accurately.⁶ But the control method depends on a better dynamic model and is unable to cope with parameter variations and model uncertainties. After BLEEX, SAC is seldom being used in other exoskeletons for its strick requirements of the model. MBC, such as direct force control⁷,⁸ and impedance control⁹,¹⁰ is relatively easy to implement. HEXAR-CR50 developed by Hanyang

¹Robotics Research Center, College of Intelligence Science and Technology, National University of Defense Technology, Changsha, China
²Hunan University of Finance and Economics, Changsha, China

Corresponding author:
Lin Lang, Hunan University of Finance and Economics, Changsha 410205, China.
Email: wuhan10@nudt.edu.cn

Creative Commons CC BY: This article is distributed under the terms of the Creative Commons Attribution 4.0 License (https://creativecommons.org/licenses/by/4.0/) which permits any use, reproduction and distribution of the work without further permission provided the original work is attributed as specified on the SAGE and Open Access pages (https://us.sagepub.com/en-us/nam/open-access-at-sage).
University uses direct force control to calculate the desired joint torques in the stance phase and then applied a proportional-integral-derivative (PID) controller at each joint.\textsuperscript{11,12} The exoskeleton developed by Massachusetts Institute of Technology (MIT) uses similar force control method under a state-machine control strategy with the support of plenty of sensor information.\textsuperscript{13} Both of them showed wearer’s metabolism reduction during locomotion.

To enhance the robustness of the exoskeleton control system, SMC\textsuperscript{14} and intelligent control NNAC\textsuperscript{15} are mainly investigated. SMC has an advantage of parameter-insensitive, and NNAC is good at approximating uncertainties. Hence, the method of SMC with NNAC compensation is widely studied as well.\textsuperscript{16–18} Though much progress has been achieved in adaptive control of load-carrying exoskeleton, few works touch on the situation of a broad range of load mass and inertia variation, which is practically unavoidable. Typical cases include loading and unloading, loads waggling during moving, impacting when weapon firing, and so on.

In this paper, a computed torque control (CTC)\textsuperscript{19} associated with fuzzy cerebellar model articulation controller (FCMAC)\textsuperscript{20,21} method is proposed for load-carrying exoskeleton on tracking problems. The biomimetic reference trajectory is given in advance since we focus on control techniques rather than motion intention detection, which is another specific research point.\textsuperscript{22} CTC, also called inverse dynamics control, is a unique application of feedback linearization of non-linear systems\textsuperscript{19} and performs well in trajectory tracking of robot manipulators.\textsuperscript{23} The CTC is made up of two feedback loops: an inner loop based on the dynamic model and an outer loop operating on the tracking error. The inner loop is used to obtain a linear and decoupled input/output relationship, whereas the outer loop is required to stabilize the overall system. To satisfy the precise model requirement of CTC, we introduced neural network compensation. FCMAC is used to approximate modeling deviation, overcome parameter variations as well as represent the dynamic payload model. This neural network is a local approximation network. For each input, only a few related weights should be adjusted, so the network can approximate non-linear terms exceptionally quickly.\textsuperscript{23–24} A clear FCMAC structure which mainly aims at significant payload variation is provided, and the unified stability analysis of the whole control scheme is proved by the Lyapunov method. Simulation results show that the proposed control method performs well during tracking even there exist model uncertainties. Moreover, the method can tolerate a broad range of payload mass over one time bigger than the mass of the device itself. With the proposed adaptive controller, exoskeletons not only help soldiers or workers transfer heavy objects but also can maintain high stability when loading and unloading without bothering operators much.

### Table 1. Parameters and variables.

| Notation | Meaning               | Unit |
|----------|-----------------------|------|
| \(m_f\)  | Foot mass             | kg   |
| \(m_s\)  | Shank mass            | kg   |
| \(m_t\)  | Thigh mass            | kg   |
| \(m_o\)  | Upper body mass       | kg   |
| \(L_f\)  | Foot length           | m    |
| \(L_s\)  | Shank length          | m    |
| \(L_t\)  | Thigh length          | m    |
| \(L_{ub}\) | Upper body length     | m    |
| \(L_{eb}\) | Length from ankle to \(G_f\) | m |
| \(L_{eb}\) | Length from ankle to \(G_s\) | m |
| \(L_{eb}\) | Length from knee to \(G_t\) | m |
| \(L_{eb}\) | Length from hip to \(G_{ub}\) | m |
| \(q_1\)  | Shank angle with respect to vertical | rad |
| \(q_2\)  | Thigh angle with respect to shank extension | rad |
| \(q_3\)  | Torso angle with respect to thigh extension | rad |
| \(\tau_1\) | Ankle torque          | Nm   |
| \(\tau_2\) | Knee torque           | Nm   |
| \(\tau_3\) | Hip torque            | Nm   |

### Dynamic model

As Figure 1 shows, exoskeleton model can be simplified as a four-bar linkage which represents foot, shank, thigh, and torso that are parallel with wearer’s limbs. \(O_{cr}, O_{cr},\) and \(O_{cr}\) are the rotation centers of the ankle, knee, and hip joint, respectively. Gravity center of each linkage is located at points \(G_f, G_s, G_t,\) and \(G_{ub}\) separately. Other parameters and variables are defined in Table 1.

As mentioned above, load-carrying exoskeleton saves wearer’s efforts by transferring heavy loads to the ground. The transfer progress only occurs in the stance phase. Thus, we mainly consider device activities in the stance phase of the sagittal plane. Figure 1 shows a state of two feet stand. A squat movement, which is a typical activity of the stance phase, is applied and researched in the following work.

The dynamic model of the stance phase exoskeleton is designed with Eurler–Lagrange equation and is expressed as

\[
M(q)\ddot{q} + C(q, \dot{q})\dot{q} + \Gamma(q) = u + J^T(q)F_{int}
\]  

(1)

where \(M\) is the inertia matrix, \(C\) is the Coriolis and centrifugal force matrix, \(\Gamma\) is the gravity matrix, \(J\) is the Jacobian matrix, \(F_{int}\) is the interaction force between exoskeleton and its wearer, and \(u = [\tau_1, \tau_2, \tau_3]^T\) is the control input torque. The interaction force \(F_{int}\) is detected by a multidimensional force sensor, which is usually mounted on the back between human and machine.\textsuperscript{25}

Note that the payload, which is usually mounted on the back of the exoskeleton torso, is not modeled considering its variation. That is, equation (1) is a non-loaded model and the existence of load will definitely influence the accuracy of \(M, C,\) and \(\Gamma.\) While those
inaccuracies can be pretty handled using the proposed control scheme. The friction term is not considered, as well. Assuming the friction is only related to the joint velocity, the existence of friction can be regarded as a deviation of $C$. Besides that, we assume the human–robot interaction force is in linear relationship with displacement errors as expressed in equation (2)

$$F_{int} = KM\Delta x + KD\Delta \dot{x} + KP\Delta x$$

If the exoskeleton tracks the reference trajectory decided by the wearer well, then the interaction force will be minimum, and that is what a load-carrying exoskeleton should achieve.

**Method**

The whole control structure is shown in Figure 2. The inputs of the exoskeleton include three parts: the physical human–robot interaction (pHRI) force $F_{int}$, the control value of CTC $u_{CTC}$, and the control quantity of FCMAC $u_{FCMAC}$. If the system model is precise enough, the CTC controller guarantees the tracking errors converging to zero by reasonable pole placement. The precise model is ensured by the FCMAC controller but cannot be ideal. Thus, a compensated controller $u_c$ is introduced.

**CTC controller**

The control law is designed as the following form

$$u = M(q)\tau + \dot{C}(\dot{q}, \ddot{q})\ddot{q} + \ddot{G}(\ddot{q}) - J^T(q)F_{int}$$

$$\tau = \ddot{q}_d + K^T E$$

$$E = \begin{bmatrix} \Delta q \\ \Delta \dot{q} \end{bmatrix}, \quad K = \begin{bmatrix} K_1 \\ K_2 \end{bmatrix}$$

where $\Delta q = q_d - q, \Delta \dot{q} = \dot{q}_d - \dot{q}, K_1, K_2 \in \mathbb{R}^{3\times3}$.

Substituting equations (3) and (4) into equation (1), the system dynamic equation is updated as

$$M(q)\ddot{q} + M(q)K^T E = M(q)\ddot{q} + \dot{C}(\dot{q}, \ddot{q})\ddot{q} + \ddot{G}(\ddot{q})$$

(6)

where $M(q) = M(q) - M(q)$, $\dot{C}(\dot{q}, \ddot{q}) = C(q, \dot{q}) - \dot{C}(\dot{q}, \ddot{q})$ and $\ddot{G}(\ddot{q}) = \ddot{G}(\ddot{q}) - \ddot{G}(\ddot{q})$. If

$$M(q)\ddot{q} + \dot{C}(\dot{q}, \ddot{q})\ddot{q} + \ddot{G}(\ddot{q}) = 0$$

(7)

Take equations (5) and (7) into equation (6), we get

$$\Delta \ddot{q} + K_2\Delta \dot{q} + K_1\Delta q = 0$$

(8)

The system will be stable as long as the coefficient matrices $K_1$ and $K_2$ are positive definite, and the position error $\Delta q$ will converge to zero. That is, the exoskeleton can follow wearer’s movement pretty well without hindering each other. However, it’s quite challenging to make equation (7) tenable because of the exists and the unpredictability of the dynamic uncertainties and external disturbances. Thus, the ideal control law equations (3) and (4) are practically limited used. To counteract those uncertainties and guarantee the tracking error $\Delta q$ converging to zero, a compensation control quantity is added based on FCMAC network.

**Architecture of FCMAC**

FCMAC architecture is similar to the conventional CMAC. The difference is that FCMAC uses Gaussian basis function as receptive field basis function rather than local constant binary function. Figure 3 shows the architecture of an n-dimensional input and p-dimensional output CMAC. The network is constructed with five spaces: input space $S$, association space $A$, receptive field space $T$, adjustable weight space $W$, and output space $Y$. And the network mainly has three mapping processes

$$R : S \Rightarrow A, \quad U : A \Rightarrow T, \quad P : T \Rightarrow Y$$

(9)

The function $\varphi = R(s)$ maps each input vector $s$ onto the association space $A$. According to $\varphi$, function $\psi = U(\varphi)$ can select specific hypercubes, which constitute the receptive field space $T$. Each hypercube corresponds with a function that is obtained by multiplying related basis functions. At last, function $y = P(\psi)$

---

![Figure 1](image1.png)  
*Figure 1. Model of the load-carrying exoskeleton in the stance phase.*

![Figure 2](image2.png)  
*Figure 2. Block diagram of the proposed control scheme.*
computes output vector $y$ by summing the weighted value of all the selected receptive field functions. The details are introduced as follows.

**Quantization.** Each data of the input vector $s$ should be quantized into a given interval to map it onto the association space conveniently. The units and ranges of the inputs may be different, but the quantization converts them into uniform dimensionless values. The minimum scale of the quantization zone decides the side length of the hypercube, which represents the size of each receptive field. Usually, we define the minimum scale as 1, and the number of scales $n_s$ is called resolution. The quantization can be linear or non-linear, and the difference has not been systematically researched. In this study, we choose $E \in \mathbb{R}^{6\times 1}$ as input that can be measured relative easily in the physical system. Resolutions of all the $n = 6$ inputs are defined the same as 9 with the minimum scale as 1. Figure 4 illustrates the quantized coordination of the inputs $s_1$ and $s_2$.

**Receptive field function.** After quantization, each input is mapped onto the association space and be represented by multilayer blocks. The association space of each input is constructed with $n_l$ layers. Each layer is divided into $n_b$ blocks. The maximum block length is defined as $l_b$. The blocks in each layer are mismatched. In general, $n_s$, $n_l$, $n_b$, and $l_b$ have a relation as shown in equation (10), so that the number of blocks in each layer is the same and the mismatch interval exactly is a minimum scale

$$n_l = l_b, \quad n_b = \frac{n_s - 1}{l_b + 1}$$

Thus, any input data can be mapped to $n_l$ blocks and each minimum scale corresponding to $n_l$ specific blocks. For each input, there are totally $n_B = n_l n_b$ blocks.

Each block corresponds with a receptive field basis function. The basis function is effected only in its related block. Any continuously bounded function can be used as the basis function, and the Gaussian function is used in this work. The basis function of the $k$th block of the $j$th layer of the $i$th input $s_i$ is presented as

$$\phi_{ijk}(s_i) = \exp \left[ \frac{-(s_i - m_{ijk})^2}{\sigma_{ijk}^2} \right]$$

Note that $k$ is related with $i$ and $j$. Any input point $s$ can be represented by a set of $n_l$ hypercubes in the receptive field. Each hypercube has a receptive field function that is composed of basic functions of the related $n$ blocks. Note that the related blocks are located in the same layer of different inputs. The $j$th layer hypercube function of the input $s$ is defined as

$$b_H(s, m, \alpha) = \prod_{i=1}^{n} \phi_{ijk} = \exp \left[ \sum_{i=1}^{n} \frac{-(s_i - m_{ijk})^2}{\sigma_{ijk}^2} \right]$$

where $m, \alpha \in \mathbb{R}^{m_B \times 1}$

$$m = [m_{11}, \ldots, m_{1n_l}, m_{21}, \ldots, m_{2n_l}, \ldots, m_{n_B1}, \ldots, m_{n_Bn_l}]^T$$

the first block the second block

the $n_B$th block

Figure 3. Architecture of CMAC.

Figure 4. Quantization and mapping of CMAC with two inputs.
\[ \sigma = [\sigma_{11}, \ldots, \sigma_{1n_l}, \sigma_{21}, \ldots, \sigma_{2n_l}, \ldots, \sigma_{n_m}, \sigma_{n_m}]^T \]  
\[ (14) \]

Define \( \Gamma \in \mathbb{R}^{n_h \times 1} \) as the vector of hypercube functions

\[ \Gamma(s, m, \sigma) = [b_{11}, \ldots, b_{1n}, b_{21}, \ldots, b_{2n}, \ldots, b_{n_m}]^T \]  
\[ (15) \]

There are \( n_h = n_l \times n_l \) hypercubes in total. Different hypercubes can be overlapped with some same receptive field blocks. We called the overlap as local generalization, which can be understood as neighboring inputs producing similar outputs. If there is no overlap between the two inputs, there is no generalization between them and they will not influence each other when the network is training.

All networks have the problem of dimensional explosion, and CMAC is no exception. As mentioned above, a CMAC with \( n \) inputs, if divided into \( n_l \) layers and each layer into \( n_h \) blocks, will have \( n_l \times n_h \) hypercubes. The amount of computation could be horrible with the increase in \( n \) and \( n_h \). However, using hash coding, the progress of mapping from association space to receptive field space can be compressed and does not influence the local generalization much at the same time.\(^{26,27}\) Thus, the computation amount and the space memory of receptive field can be reduced to an acceptable range, \( n_h = n_B \), for example.\(^{28}\)

**Weight space and output.** Each hypercube corresponds to a specific weight when mapping to a specific output. Suppose the output dimension is \( p \), the weight space \( W \) is \( p \) times large as the receptive field space \( T \)

\[ w = [w_1, w_2, \ldots, w_p] = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1p} \\
1 & 2 & \cdots & 12
\end{bmatrix} \]  
\[ (16) \]

in which \( w_i (i \in 1, 2, \cdots, p) \) is the weight vector of the \( i \)th output \( y_i \). The output \( y \in \mathbb{R}^{n_h \times p} \) is expressed as

\[ y = [y_1, y_2, \cdots, y_p]^T = w^T \Gamma(s, m, \sigma) \]  
\[ (17) \]

For example, if two input vectors are quantized to (2.1, 2.4) and (2.5, 2.6), respectively, they are both located at the square (3,3) and represented by the hypercubes \( ^1C^2C, ^1D^2D, ^1E^2E, \) and \( ^1F^2F \).

**FCMAC and compensated controller**

Redesign the control law equation (3) as

\[ u = M \tau + \tilde{C} \tilde{q} + \tilde{G} - \tilde{J}^T(q) F_{in} + u_{\text{FCMAC}} \]  
\[ (18) \]

Then equation (6) can be rewritten as

\[ \Delta \dot{q} = - K^T E + M(q)^{-1} \Psi \]  
\[ (19) \]

in which \( \Psi \) represents the uncertainties and the neural network output term

\[ \Psi = \tilde{M}(q) \dot{q} + \tilde{C}(q, \dot{q}) \dot{q} + \tilde{G} - u_{\text{FCMAC}} \]  
\[ (20) \]

The system state-space dynamic model can be written as

\[ \dot{E} = AE + B \Psi \]  
\[ (21) \]

where

\[ E = [\Delta \dot{q} \Delta q]^T, \quad A = \begin{bmatrix} 0 & I^{3 \times 3} \\
-\dot{K}_1 & -\dot{K}_2 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\
\tilde{F}^{3 \times 3} \end{bmatrix} \]

If there exists ideal weight values \( w^* \) and parameters \( m^*, \sigma^* \) that can make \( \Psi \to 0 \), then the uncertainties can be represented as

\[ u_{\text{FCMAC}} = \tilde{M}(q) \dot{q} + \tilde{C}(q, \dot{q}) \dot{q} + \tilde{G} = w^T \tilde{\Gamma}(s, m^*, \sigma^*) + \varepsilon(s) \]  
\[ (22) \]

in which \( \varepsilon(s) \) is the functional reconstructional error and \( || \varepsilon(s) || = \varepsilon_M \) is bounded.\(^{29}\) Let \( \tilde{w}, \tilde{m}, \) and \( \tilde{\sigma} \) are the estimated values of \( w^*, m^* \), and \( \sigma^* \), the output of FCMAC network \( u_f \) is

\[ u_f = \tilde{w}^T \tilde{\Gamma}(s, \tilde{m}, \tilde{\sigma}) \]  
\[ (23) \]

To minimize the approximation error of the network, a compensation quantity \( u_c \) is added. The approximation error \( \tilde{u} \) is expressed as

\[ \tilde{u} = u^* - u_{\text{FCMAC}} = \tilde{w}^T \tilde{\Gamma} + \tilde{w}^T \tilde{\Gamma} + \varepsilon - u_c \]  
\[ (24) \]

where \( \tilde{w} = w^* - \tilde{w} \) and \( \tilde{\Gamma} = \Gamma^* - \Gamma \). Linearize \( \tilde{\Gamma} \) using Taylor expansion

\[ \tilde{\Gamma} = \begin{bmatrix} \frac{\partial b_1}{\partial m} & \frac{\partial b_2}{\partial m} & \cdots & \frac{\partial b_n}{\partial m} \\
\frac{\partial b_1}{\partial \sigma} & \frac{\partial b_2}{\partial \sigma} & \cdots & \frac{\partial b_n}{\partial \sigma} \end{bmatrix} \]  
\[ (m^* - \tilde{m}) + \frac{\partial b_k}{\partial \sigma} (\sigma^* - \tilde{\sigma}) + O_t \]  
\[ = G^* \tilde{m} + H^T \tilde{\sigma} + O_t \]  
\[ (25) \]

where \( O_t \in \mathbb{R}^{m \times 1} \) is a higher-order term and \( G, H \in \mathbb{R}^{m_m \times n_h} \) are the coefficient matrices with elements expressed as (symbol “\( \sim \)” means the block left of the symbol is one of the elements that construct the hypercube right of the symbol)

\[ \frac{\partial b_k}{\partial m} = \begin{cases} \frac{2(\dot{m} - m_0)}{m_k^2} b_k(s, m, \sigma), & \text{if } \phi_{ij} \sim b_k \\
0, & \text{else} \end{cases} \]  
\[ (26) \]

\[ \frac{\partial b_k}{\partial \sigma} = \begin{cases} \frac{2(\dot{\sigma} - \sigma_0)}{\sigma_k^2} b_k(s, m, \sigma), & \text{if } \phi_{ij} \sim b_k \\
0, & \text{else} \end{cases} \]  
\[ (27) \]

The optimal hypercube function set is represented as

\[ \Gamma^* = \tilde{\Gamma} = \tilde{\Gamma} = G^* \tilde{m} + H^T \tilde{\sigma} + O_t \]  
\[ (28) \]
Substitute equations (20), (22), (24), and (28) to equation (21)

\[
\dot{E} = AE + B\ddot{u} \\
= AE + B(\dot{w}^T\Gamma + \dot{w}^T\Gamma + \dot{w}^T\Gamma + \varepsilon - u_e) \\
= AE + B(\dot{w}^T\Gamma + \dot{w}^T\Gamma + \dot{w}^T\Gamma + \varepsilon - u_e) \\
= AE + B\left[\dot{w}^T\Gamma + \dot{w}^T(G^Tm + H^T\dot{\sigma}) + \eta - u_e\right]
\]  
(29)

in which \(\eta\) is the error of the network and is bounded \(\eta_0\). The FCMAC and the compensated control laws are directly given using the Lyapunov method

\[
\dot{\tilde{w}} = \gamma_1 E^T PB \\
\dot{\tilde{m}} = \gamma_2 G\tilde{w}^T PE \\
\dot{\tilde{\sigma}} = \gamma_3 H\tilde{w}^T PE \\
u_c = \tilde{\eta}, \quad \eta = 4E^T PB
\]  
(30) (31) (32) (33)

where \(\gamma_1, \gamma_2, \gamma_3,\) and \(\gamma_4\) are some constants greater than zero. \(P\) is a symmetric positive definite matrix and fulfills the following Lyapunov equation

\[
PA + A^TP = -Q
\]  
(34)

and \(Q\) is a positive definite matrix.

As for the load-carrying exoskeleton described in equation (1), the control laws, which correspond with the control scheme in Figure 2, are entirely designed as

\[
u = u_{CTC} + u_{CMAC} = u_{CTC} + uf + u_e
\]  
(35)

in which \(u_{CTC}\) is designed as equations (3)–(5), \(uf\) is expressed in equation (23) with parameters presented in equations (30)–(32), and \(u_e\) is defined in equation (33).

**Stability analysis**

Define the Lyapunov function as

\[
V(E, \dot{w}, \tilde{m}, \dot{\sigma}, t) = \frac{1}{2} E^T PE + \frac{1}{2\gamma_1} \dot{E} \| \dot{w} \|^2 P + \frac{1}{2\gamma_2} \dot{\tilde{m}}^T \tilde{m} \\
+ \frac{1}{2\gamma_3} \dot{\tilde{\sigma}}^T \tilde{\sigma} + \frac{1}{2\gamma_4} \dot{\eta}^T \eta
\]  
(36)

Taking the derivative of the Lyapunov function and substituting equations (29) and (34) successively, we get

\[
\dot{V} = \frac{1}{2} E^T PE + \frac{1}{2} E^T PE - \frac{1}{\gamma_1} tr(\dot{w}^T \dot{w}) - \frac{1}{\gamma_2} \dot{\tilde{m}}^T \dot{\tilde{m}} \\
- \frac{1}{\gamma_3} \dot{\tilde{\sigma}}^T \dot{\tilde{\sigma}} - \frac{1}{\gamma_4} \dot{\eta}^T \dot{\eta}
\]  
(37)

According to equations (30)–(33), equation (37) is simplified as

\[
\dot{V} = -\frac{1}{2} E^T Q E + 0 + 0 + 0 = -\frac{1}{2} E^T Q E \leq 0
\]  
(38)

where \(\dot{V}\) is zero only occurs at the situation of \(E = 0\). According to Barbalat’s Lemma, \(E \to 0\) as \(t \to \infty\).

**Simulation**

The proposed method is verified through simulations. The leg on the stance phase rather than the swing phase is selected as the application object because the load variation has a slight influence on the swing leg. A squat movement is used in the simulation. The desired trajectories of the ankle, knee, and hip joints are obtained through the fitting results of human squatting data, which are collected by the motion analysis equipment. Figure 5 is the squatting progress during the simulation.
The average mass $m = 25$ kg of the existing exoskeleton products is chosen as the total weight of the load-carrying exoskeleton simulation model. The parameter values in Table 1 are specified as $m_f = 3.0$ kg, $m_s = 6.0$ kg, $m_t = 6.0$ kg, $m_{ub} = 10.0$ kg, $L_f = 270$ mm, $L_s = 520$ mm, $L_t = 450$ mm, and $L_{ub} = 530$ mm.

Assume all the links are homogeneous, and the mass point is located at the geometrical center.

As for FCMAC network, joint angle displacements and their differential $E$ are chosen as the inputs and the outputs $u_{CMAC}$ are the joint torque control quantity. Assume the three joint angle displacements $\Delta \theta$ have a range of $[-\pi/90, \pi/90]$, and the differential of angular deviations $\Delta \dot{\theta}$ changes within the range of $[-\pi/15, \pi/15]$. All of the six inputs are quantized to the section $[0, 9]$ with a minimum scale of 1. The block length $b_l$ is chosen as 4. Hence, there are four layers for each input in the association space, and each layer has three blocks. The mean values of the basis function $m$ are initialized as the center value of each block and the standard deviations $\sigma$ as zero. The weight $w$ is initialized as zero as well.

Three groups of simulation in different conditions are designed in this paper. The first is without model uncertainties. The second is with model parameter changes, and the last is with load changes.

### Simulation without uncertainties

In this condition, no load is added on the exoskeleton. The dynamic system model is definite and is used in the CTC controller design. The parameters used in the controller are designed as $K_1 = 600I^3$, $K_2 = 100I^3$, $Q = 100I^6$, $\gamma_1 = 2000$, $\gamma_2 = 100$, $\gamma_3 = 100$, and $\gamma_4 = 10$. Figure 6 is the trajectory tracking results during a period of squatting. The system tracks the desired trajectory well whether using the neural network or not, which indicates that the CTC controller is adequate. Moreover, from the tracking errors, we can see the CMAC controller can restrain errors and plays a positive role during tracking.

### Simulation with model parameter changes

The model parameter changes come from the modeling simplification and the measuring errors. In this condition, the load is not considered for the moment. For simplicity, we set the model parameter changing by adjusting the value of $M$, $C$, and $G$ in equation (1). The maximum variable quantity is ±40%. The controllers are designed identical as in the above subsection. The tracking results are displayed in Figures 7 and 8.

In the condition that the model parameters increase 40%, the maximum tracking errors of the control without FCMAC and with FCMAC are $2.12 \times 10^{-2}$ and $1.8 \times 10^{-3}$ rad, respectively. Both of the maximum errors occur at the knee joint. Favorable tracking response is obtained when using the FCMAC controller. When the model parameters decrease 40%, a similar control effect appears that the maximum tracking errors of control without FCMAC and with FCMAC are $3.84 \times 10^{-2}$ and $1.6 \times 10^{-3}$ rad, respectively. From the above two simulations, it can be seen that without the FCMAC controller, though the joints can track the desired trajectory, the tracking errors are relatively high, which increases the interaction force between the pilot and exoskeleton. However, with the FCMAC controller, more precise tracking responses can be guaranteed.

### Simulation with load changes

The load adheres to the exoskeleton torso. During the working, the load supported by the exoskeleton can be changed frequently; for example, loading and unloading equipment, weapon launching, fuel consumption, and so on. Thus the control system should have the
ability to cope with load variations. In this condition, only load mass variation is considered, and the load position and orientation variations are not taken into account, which can be regarded as the model parameter changes in the above subsection.

The initial state of the system used in the simulation is non-loaded. Three groups of simulation are designed with 10, 20, and 30 kg load, respectively. The controllers stay the same as subsection “Simulation without Uncertainties.” The tracking responses are illustrated in Figure 9. For comparison, the tracking responses without FCMAC controller are displayed in Figure 10.

Along with the adding of load, the tracking error of both the controls with FCMAC and without FCMAC is increasing. However, the control without FCMAC is hard to track the desired trajectory, especially when the load is heavy. The maximum tracking errors of the three joints under the condition of 30 kg load in Figure 10 are 0.05, 0.21 and −0.33 rad, respectively. While using the control with FCMAC, the maximum tracking errors are reduced to \(-1.3 \times 10^{-3}, 2.6 \times 10^{-3}\) and \(1.7 \times 10^{-3}\) rad, respectively. The proposed control scheme can well deal with the load variation.

**Figure 7.** Squatting control with model parameters increase 40%. (a) trajectory tracking and (b) trajectory errors.

**Figure 8.** Squatting control with model parameters decrease 40%. (a) Trajectory tracking. (b) Trajectory errors.

**Conclusion**

A trajectory tracking controller associated with FCMAC compensation for load-carrying exoskeleton in the stance phase is proposed in this paper. The control scheme is mainly composed of CTC and FCMAC. The CTC is the primary controller for the trajectory tracking and has a benefit of smooth control quantity compared with the sliding mode controller. The FCMAC is used to approximate the dynamic model uncertainties. The system stability is proved by the Lyapunov function. The simulation results show that the proposed control strategy has a perfect tracking response in squatting movements and can better deal with situations, such as model uncertainties and load variations.

The proposed controller has a robust ability of system uncertainties. As to the flexibility of the load-carrying exoskeleton, it can be affected by many factors besides the robust control. First, the flexibility is related with the load weight and the motion state. In practical applications, the power of the actuators of the exoskeleton is limited. Second, the load-carrying exoskeleton can be regarded as a biped robot. During walking, the
flexibility of the exoskeleton is mainly affected by the swing phase and the swing–stance–switch phase. Then, if the system uncertainties are too large, the proposed control method cannot guarantee the tracking effect. Because the higher-order term $O_t$ in equation (25) will have large impact on the controller. Also, in this paper, the desired trajectory is given directly. Actually, the acquisition of the desired trajectory is a challenge of intention perception of human motion. The accuracy of motion intention perception decides the control effect of the exoskeleton. So, if the power of the actuators is large enough, the system uncertainties are limited (the load is less than 30 kg and the errors of the model parameters are below 40%) and the desired trajectory is accuracy enough, the proposed FCMAC-based adaptive tracking control impacts the flexibility of the exoskeleton less.

Future work includes designing proper association space of FCMAC, researching non-linear quantization of FCMAC, and simulating the control strategy on various movements. Considering that the proposed control strategy has a high tolerance of uncertainties, the swing leg of the exoskeleton can be regarded as disturbances of the stance phase model, thus the control strategy of the complex activities can be simplified. The method will be evaluated on a physical system in the future.

Declaration of conflicting interests
The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding
The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This research was funded by the National Key R&D Program of China, Grant No. 2018YFC2001304.

ORCID iD
Han Wu https://orcid.org/0000-0002-4376-5391

References
1. Herr H. Exoskeletons and orthoses: classification, design challenges and future directions. J Neuroeng Rehabil 2009; 6(1): 21.
1. Pons JL. Wearable robots: biomechatronic exoskeletons. Hoboken, NJ: John Wiley & Sons, 2008.
2. Huo W, Mohammed S, Moreno JC, et al. Lower limb wearable robots for assistance and rehabilitation: a state of the art. *IEEE Syst J* 2014; 10(3): 1068–1081.
3. Zoss AB, Kazerooni H and Chu A. Biomechanical design of the Berkeley lower extremity exoskeleton (BLEEX). *IEEE/ASME T Mech* 2006; 11(2): 128–138.
4. Cornwell W. In pursuit of the perfect power suit. *Science* 2015; 350: 270–273.
5. Steger R, Kim SH and Kazerooni H. Control scheme and networked control architecture for the Berkeley lower extremity exoskeleton (BLEEX). In: *Proceedings 2006 IEEE international conference on robotics and automation*, Orlando, FL, 15–19 May 2006, pp. 3469–3476. New York: IEEE.
6. Zhu Y, Zhang G, Zhang C, et al. Biomechanical modeling and load-carrying simulation of lower limb exoskeleton. *Biomed Mater Eng* 2015; 26(S1): S729–S738.
7. Sahin Y, Botsali FM, Kalyoncu M, et al. Force feedback control of lower extremity exoskeleton assisting of load carrying human. *Appl Mech Mater* 2014; 598: 546–550.
8. Aguirre-Ollinger G, Colgate JE, Peshkin MA, et al. Active-impedance control of a lower-limb assistive exoskeleton. In: *Proceedings of the 2007 IEEE 10th international conference on rehabilitation robotics*, Noordwijk, 13–15 June 2007, pp. 188–195. New York: IEEE.
9. Yang Z, Xu S, Zhuo B, et al. Human-robot cooperation based on dynamic control of upper limb exoskeleton for human power amplification. *Mechatronics* 2014; 24(2): 168–176.
10. Walsh CJ, Pasch K and Herr H. An autonomous, underactuated exoskeleton for load-carrying augmentation. In: *Proceedings of the 2006 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, Hamburg, 28 September–2 October 2015, pp. 5345–5350. New York: IEEE.
11. Lee HD, Lee BK, Kim WS, et al. Human-robot cooperation based on a dynamic model of an upper limb exoskeleton for human power amplification. *IEEE Trans Robot* 2014; 30(1): 22–31.