

Research Article

Robust Sliding Mode Control Based on GA Optimization and CMAC Compensation for Lower Limb Exoskeleton

Yi Long, Zhi-jiang Du, Wei-dong Wang, and Wei Dong

State Key Laboratory of Robotics and System, Harbin Institute of Technology, Harbin 150001, China

Correspondence should be addressed to Wei Dong; dongwei@hit.edu.cn

Received 16 January 2016; Accepted 17 February 2016

Academic Editor: Huapeng Wu

Copyright © 2016 Yi Long et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A lower limb assistive exoskeleton is designed to help operators walk or carry payloads. The exoskeleton is required to shadow human motion intent accurately and compliantly to prevent incoordination. If the user's intention is estimated accurately, a precise position control strategy will improve collaboration between the user and the exoskeleton. In this paper, a hybrid position control scheme, combining sliding mode control (SMC) with a cerebellar model articulation controller (CMAC) neural network, is proposed to control the exoskeleton to react appropriately to human motion intent. A genetic algorithm (GA) is utilized to determine the optimal sliding surface and the sliding control law to improve performance of SMC. The proposed control strategy (SMC_GA_CMAL) is compared with three other types of approaches, that is, conventional SMC without optimization, optimal SMC with GA (SMC_GA), and SMC with CMAC compensation (SMC_CMAL), all of which are employed to track the desired joint angular position which is deduced from Clinical Gait Analysis (CGA) data. Position tracking performance is investigated with cosimulation using ADAMS and MATLAB/SIMULINK in two cases, of which the first case is without disturbances while the second case is with a bounded disturbance. The cosimulation results show the effectiveness of the proposed control strategy which can be employed in similar exoskeleton systems.

1. Introduction

The lower extremity exoskeleton, which began in the late 1960s, is an electromechanical structure worn by human users as an intelligent device for performance assistance and enhancement. In recent years, wearable robots have attracted interests of many researchers widely. The Berkeley Lower Extremity Exoskeleton (BLEEX) was designed to assist people in walking for carrying load, which could walk at the speed of 0.9 m/s while carrying 34 kg payload [1]. A mechanical leg has seven DOFs (three at the hip, one at the knee, and three at the ankle), of which four DOFs are actuated by valve-based hydraulic actuation systems [2]. However, these many active DOFs make the system complex and heavy, weighing 38 kg. The latter exoskeletons, that is, ExoHiker, ExoClimber, and HULC, simplify mechanical structure and reduce the number of active DOFs while carrying more payloads up to 68 kg–90 kg [3]. Hybrid Assistive Limb (HAL), proposed by the University of Tsukuba in Japan, has two active DOFs at the hip joint and knee joint, which are controlled according to

collected electrical signals from muscles [4]. HAL is used to help users carry load and assist disabled people in walking [5, 6]. An underactuated exoskeleton system is designed based on appropriate criteria to help infantry soldiers walk on different terrain, where active joints are applied to the knee joints while other joints are passive [7]. Moreno et al. studied and analyzed the human interaction with wearable lower limb exoskeleton, where the robot gathered information from the sensors in order to detect human actions and subjects also modified their gait patterns to obtain the desired responses from the exoskeleton [8].

Although many kinds of lower limb exoskeleton robots are studied, the human-exoskeleton collaborative movement is quite complex and difficult due to nonlinear characteristics of dynamic model and uncertainties, for example, external disturbance and involuntary movements. To achieve the goal of making exoskeletons providing assistance for human beings, a consistent dynamic tracking performance is required to maneuver exoskeletons in an efficient, smooth, and continuous manner [9]. The control procedure can be

divided into two steps, acquiring human motion intent with human-robot interaction (HRI) and following the human motion intent accurately.

When the wearer wants to move, the central controller sends control signals to enforce the exoskeleton to follow commanded signals, during which HRI decreases. A crucial issue of control is to follow the estimated human motion intent accurately. The more accurate the intention tracking is, the more compliantly the exoskeleton works. The precise motion control of robotic manipulators has received considerable attention from many robotics researchers and its challenges continue to limit overall control performance because of structured and unstructured uncertainties [10]. In exoskeletons, the structured uncertainties contain payload variations, while unstructured uncertainties contain sensor noises, joint friction, and external disturbances. There are many approaches for position control approaches to deal with uncertainties such as robust control [11, 12], adaptive control [13, 14], intelligent control [15], and sliding mode control [16].

SMC is a robust control approach that drives state trajectory to predefined sliding surface by using discontinuous control inputs [17], which is used to improve control performance for robotic manipulators with model uncertainties such as parameter perturbations, unknown joint frictions and inertias, and external disturbances [18]. It is notable that its overall performance is superior to general PID control algorithm [19]. The process of designing a SMC controller has two steps: defining suitable sliding surfaces and designing discontinuous control laws [13]. Parameters of SMC should be chosen suitably to obtain optimal performance. Some common optimization methods are provided and applied in robots, for example, GA [20], particle swarm optimization (PSO) [21], ant colony optimization (ACO) [22], and evolutionary algorithm (EA) [23]. GA is simple to be implemented and is capable of locating global optimal solutions [24], which is utilized to optimize the structure of intelligent methods [25, 26]. The decoupled SMC as a supervisory controller is applied in accordance with PID control, whose parameters are tuned using GA, to enhance tracking performance and eliminate the chattering problem [27]. The gain switch and sliding surface constant parameters are selected by GA so that the designed SMC can achieve satisfactory performance [28]. However, GA is only used to optimize parameters of sliding surfaces or SMC control laws. In this work, we use GA to optimize all parameters of the sliding surface and the control law at the same time.

The optimal SMC can deal with uncertainties to achieve satisfactory performance. To improve tracking performance, CMAC is added as a compensation item with property of fast learning capability. The CMAC proposed first by Albus [29] is similar to the mode of human cerebellum, which is an autoassociative memory feed-forward neural network. Compared with other feed-forward neural networks, it has faster convergence speed [30]. The approach which uses CMAC as a compensation item with SMC is applied in position control of robotic manipulators [31]. In this work, we propose to combine optimal SMC using GA and CMAC compensation to form the hybrid position control strategy.

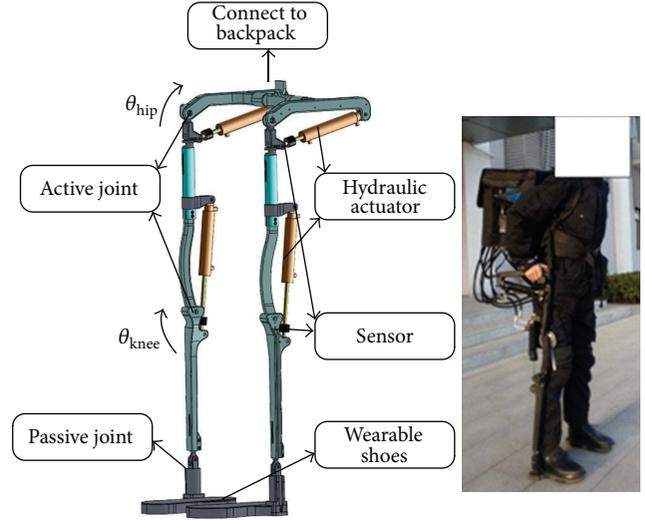


FIGURE 1: Prototype of lower limb powered exoskeleton. There are two active joints of each leg in walking direction, which are represented as θ_{hip} and θ_{knee} . All auxiliary facilities are packaged in the backpack.

The remainder of this paper is organized as follows. The specific system under study is given in the second section. In Section 3, the proposed control strategy is explained in details. Cosimulations using the proposed approach and results analysis are presented in the fourth section. Conclusions are drawn in the final section.

2. Problem Formulations

2.1. Exoskeleton Configuration. Based on principles in biological design, the designed exoskeleton is required to retain adaptability to multifunctionality of human lower limbs. An available powerful tool when designing an assistive exoskeleton is the enormous Clinical Gait Analysis (CGA) data on human walking [32]. With CGA data [33], our designed exoskeleton is shown in Figure 1. As Figure 1 shows, there are two active joints of a single leg in sagittal plane, which are knee joint and hip joint actuated by hydraulic actuation system.

2.2. Mathematical Model of Exoskeleton. For multirigid system, Euler-Lagrange is a frequently used method for modeling of robotic manipulators. The exoskeleton is a typical human-robot collaboration system, which includes the user's lower limbs and mechanical limbs which are tied together at the interaction cuffs. Mathematical model of a single leg of exoskeleton is obtained because of its symmetry structure. Without loss of generality, the dynamic equation of the swing leg of exoskeleton robot can be expressed as follows:

$$\mathbf{M}(\mathbf{q}) \ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) + \mathbf{D} = \mathbf{T}, \quad (1)$$

where $\mathbf{M}(\mathbf{q}) \in R^{n \times n}$ is the symmetric definite inertial matrix; $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \in R^{n \times n}$ is the Coriolis and centrifugal force matrix; $\mathbf{G}(\mathbf{q}) \in R^{n \times 1}$ is the gravitational force matrix; $\mathbf{T} \in R^{n \times 1}$

is the control input vector; $\mathbf{D} \in R^{n \times 1}$ denotes unmodeled dynamics and external disturbances.

For dynamics model in (1), several properties are presented as the following [34].

Property 1. Matrix $\mathbf{M}(\mathbf{q})$ is symmetric and positive definite.

Property 2. Matrix $\dot{\mathbf{M}}(\mathbf{q}) - 2\mathbf{C}(q, \dot{q})$ is a skew-symmetric matrix if $\forall \varepsilon \in R^n, \varepsilon^T(\dot{\mathbf{M}}(\mathbf{q}) - 2\mathbf{C}(q, \dot{q}))\varepsilon = 0$.

Property 3. There exist finite scalars $\delta_i > 0, i = 1, \dots, 4$ such that $\|\mathbf{M}(\mathbf{q})\| \leq \delta_1, \|\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\| \leq \delta_2, \|\mathbf{G}(\mathbf{q})\| \leq \delta_3$, and $\|D\| \leq \delta_4$, which means all items in dynamic model are bounded.

In the position control of robotic manipulators, we define trajectory tracking error as

$$\mathbf{e} = \mathbf{q}_d - \mathbf{q}, \quad (2)$$

where \mathbf{e} is the tracking error, \mathbf{q}_d is reference trajectory, and \mathbf{q} is actual trajectory. Based on (2), we can obtain

$$\begin{aligned} \dot{\mathbf{e}} &= \dot{\mathbf{q}}_d - \dot{\mathbf{q}}, \\ \ddot{\mathbf{e}} &= \ddot{\mathbf{q}}_d - \ddot{\mathbf{q}}, \end{aligned} \quad (3)$$

where $\dot{\mathbf{e}}$ and $\ddot{\mathbf{e}}$ is the first and second derivative of \mathbf{e} , $\dot{\mathbf{q}}_d$ and $\ddot{\mathbf{q}}_d$ are angular velocity and acceleration vector of command input, and $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$ are that of actual output, respectively, all of which are bounded.

3. Control Strategy Design

3.1. Sliding Mode Control. A general SMC design consists of two steps: the sliding surface design and the control law construction. The purpose of the SMC is to track the trajectory specified by human intention and maintain system trajectory in the sliding surfaces [18]. Considering that there exist uncertainties including unmodeled frictions, variation of parameters, and external disturbances, the robustness should be an important concern in the controller design for exoskeleton system. The general sliding surface is defined as $\mathbf{s} = \dot{\mathbf{e}} + \mathbf{A}\mathbf{e}$. To improve robustness of controller, a designed integral sliding surface is represented as follows [35]:

$$\mathbf{s} = \dot{\mathbf{e}} + \mathbf{A}\mathbf{e} + \mathbf{H} \int_0^{ts} \mathbf{e} dt, \quad (4)$$

where \mathbf{A} and \mathbf{H} are positive definite matrix. Then $\dot{\mathbf{s}}$ can be derived:

$$\dot{\mathbf{s}} = \ddot{\mathbf{e}} + \mathbf{A}\dot{\mathbf{e}} + \mathbf{H}\mathbf{e}. \quad (5)$$

As the second design stage of SMC, the control laws should be chosen, which should be satisfied with the existence condition of SMC [36]:

$$\mathbf{s}^T \dot{\mathbf{s}} < 0. \quad (6)$$

For the exoskeleton system under study, we define the SMC control law as follows:

$$\begin{aligned} \mathbf{u} &= \mathbf{M}(q) \ddot{\mathbf{q}}_d - (\mathbf{T}_d - \mathbf{C}(q, \dot{q}) \dot{\mathbf{q}}) + \mathbf{M}(q) \mathbf{A}\dot{\mathbf{e}} \\ &+ \mathbf{M}(q) \mathbf{H}\mathbf{e} + \mathbf{C}(q, \dot{q}) \mathbf{s} + \varepsilon \operatorname{sgn}(\mathbf{s}) + \mathbf{K}\mathbf{s}, \end{aligned} \quad (7)$$

where $\mathbf{T}_d = \mathbf{D} - \mathbf{G}(\mathbf{q})$, ε and \mathbf{K} are positive definite matrices, and $\operatorname{sgn}(\mathbf{s})$ is a symbolic function which is shown as follows:

$$\operatorname{sgn}(\mathbf{s}) = \begin{cases} 1, & \mathbf{s} > 0, \\ 0, & \mathbf{s} = 0, \\ -1, & \mathbf{s} < 0. \end{cases} \quad (8)$$

The SMC algorithm has chattering phenomena, which affects the accuracy of position control much. In order to eliminate chattering, the continuous function $\theta(\mathbf{s})$ with relay characteristics is used to replace the function of symbolic function $\operatorname{sgn}(\mathbf{s})$ to restrict the trajectory in a boundary layer of ideal sliding mode [37]. Then (7) can be rewritten as

$$\begin{aligned} \mathbf{u} &= \mathbf{M}(q) \ddot{\mathbf{q}}_d - (\mathbf{T}_d - \mathbf{C}(q, \dot{q}) \dot{\mathbf{q}}) + \mathbf{M}(q) \mathbf{A}\dot{\mathbf{e}} \\ &+ \mathbf{M}(q) \mathbf{H}\mathbf{e} + \mathbf{C}(q, \dot{q}) \mathbf{s} + \varepsilon \theta(\mathbf{s}) + \mathbf{K}\mathbf{s}, \end{aligned} \quad (9)$$

where $\theta(\mathbf{s}) = \mathbf{s}/(\|\mathbf{s}\| + \sigma)$, $\sigma > 0$. Before stability analysis, *Barbalat lemma* is shown as the following [38].

Barbalat Lemma. *If a differentiable function $f(t)$ has a limit as $t \rightarrow \infty$, and if $\dot{f}(t)$ is uniformly continuous, then $f(t) \rightarrow 0$ as $t \rightarrow \infty$.*

Theorem 1. *The proposed controller (9) guarantees asymptotic convergence to zero, both of the trajectory tracking errors and sliding surfaces. Namely, the system is globally stable; that is, when $t \rightarrow \infty, e \rightarrow 0, s \rightarrow 0$.*

Proof. Lyapunov function is defined as

$$\mathbf{V} = \frac{1}{2} \mathbf{s}^T \mathbf{M}(q) \mathbf{s}. \quad (10)$$

Differentiating \mathbf{V} with respect to time yields

$$\dot{\mathbf{V}} = \mathbf{s}^T \mathbf{M}(q) \dot{\mathbf{s}} + \frac{1}{2} \mathbf{s}^T \dot{\mathbf{M}}(q) \mathbf{s}. \quad (11)$$

Considering Property 2, then

$$\mathbf{s}^T \left(\frac{1}{2} \dot{\mathbf{M}}(q) - \mathbf{C}(q, \dot{q}) \right) \mathbf{s} = 0. \quad (12)$$

Combining (10)–(12), one can get

$$\begin{aligned} \dot{\mathbf{V}} &= \mathbf{s}^T (\mathbf{M}(q) \dot{\mathbf{s}} + \mathbf{C}(q, \dot{q}) \mathbf{s}) = \mathbf{s}^T (\mathbf{M}(q) (\ddot{\mathbf{q}}_d - \ddot{\mathbf{q}}) \\ &+ \mathbf{M}(q) \mathbf{A}\dot{\mathbf{e}} + \mathbf{M}(q) \mathbf{H}\mathbf{e} + \mathbf{C}(q, \dot{q}) \mathbf{s}). \end{aligned} \quad (13)$$

And $\ddot{\mathbf{q}}$ can be solved by

$$\ddot{\mathbf{q}} = \mathbf{M}(q)^{-1} (\mathbf{T} + \mathbf{T}_d - \mathbf{C}(q, \dot{q}) \dot{\mathbf{q}}). \quad (14)$$

Substituting (14) into (13), then

$$\begin{aligned} \dot{\mathbf{V}} &= \mathbf{s}^T (\mathbf{M}(q) \ddot{\mathbf{q}}_d - (\mathbf{T} + \mathbf{T}_d - \mathbf{C}(q, \dot{q}) \dot{\mathbf{q}}) + \mathbf{M}(q) \mathbf{A}\dot{\mathbf{e}} \\ &+ \mathbf{M}(q) \mathbf{H}\mathbf{e} + \mathbf{C}(q, \dot{q}) \mathbf{s}). \end{aligned} \quad (15)$$

Substituting (7) into (15), we can obtain

$$\dot{\mathbf{V}} = \mathbf{s}^T (-\boldsymbol{\varepsilon}\theta(\mathbf{s}) - \mathbf{K}\mathbf{s}). \quad (16)$$

It is easy to know that \mathbf{K} and $\boldsymbol{\varepsilon}$ are positive definite matrices; therefore $\mathbf{s}^T \mathbf{K}\mathbf{s} > 0$, $\mathbf{s}^T \theta(\mathbf{s}) > 0$; then $\dot{\mathbf{V}} < 0$. Hence, the system is globally stable. With the *Barbalat lemma*, $\mathbf{s} \rightarrow 0$ as $t \rightarrow \infty$; then one knows $\mathbf{e} \rightarrow 0$ and $\dot{\mathbf{e}} \rightarrow 0$ as $t \rightarrow \infty$. This control law could realize convergence of the trajectory tracking error to zero. \square

3.2. Genetic Algorithm. In SMC, those constant parameters existing in sliding surfaces and control laws, which are \mathbf{A} , \mathbf{H} , \mathbf{K} , and $\boldsymbol{\varepsilon}$ in (9), determine the overall performance. Hence, it is necessary to find the optimal values of them using optimization algorithm. GA is an adaptive heuristic search algorithm that mimics the process of natural selection and uses biological evolution to develop a series of search space points toward an optimal solution. There are five components that are required to implement GA: representation, initialization, fitness function, genetic operators, and genetic parameters [39].

A simple GA involves three types of operator: selection, crossover, and mutation [40]. Selection is a probabilistic process for selecting chromosomes in the population using their fitness values. The chromosome with larger fitness value is likely to be selected to reproduce. Crossover is the process of randomly choosing a locus and swaps the characters either left or right of this locus between two chromosomes to create two offspring. The probability of crossover occurring for the parent chromosomes is usually set to a large value (e.g., 0.8). Mutation is to randomly flip some of the bits by changing “0” to “1” or vice versa, with a small probability (e.g., 0.001) which maintains genetic diversity to guarantee that GA can come to better solution. The process of GA optimization is shown in Figure 2. As Figure 2 shows, there are parameters such as the size of population and generation and the length of code that should be initialized; then the process of selection, crossover, and mutation is preceded until the convergence conditions are satisfied.

3.3. SMC with GA Optimization. Based on that discussed above, the fitness function should be confirmed before implementing GA to SMC. The goal of SMC is to achieve precise trajectory tracking for robotic manipulators; that is, the smaller the trajectory errors are, the more effective the controller is. Those parameters to be optimized are relevant to trajectory error; hence the fitness function is defined as follows:

$$J(\mathbf{A}, \mathbf{H}, \mathbf{K}, \boldsymbol{\varepsilon}) = \sum_{k=0}^{\infty} \|\mathbf{e}(k)\|^2. \quad (17)$$

With the fitness function, those parameters can be found with the minimization of tracking errors during the trajectory tracking using the designed control law. In the search space of GA, the SMC will have optimal parameters when the fitness function has minimum values. The algorithm of SMC optimized by GA is shown as Algorithm 1 in Appendix A.

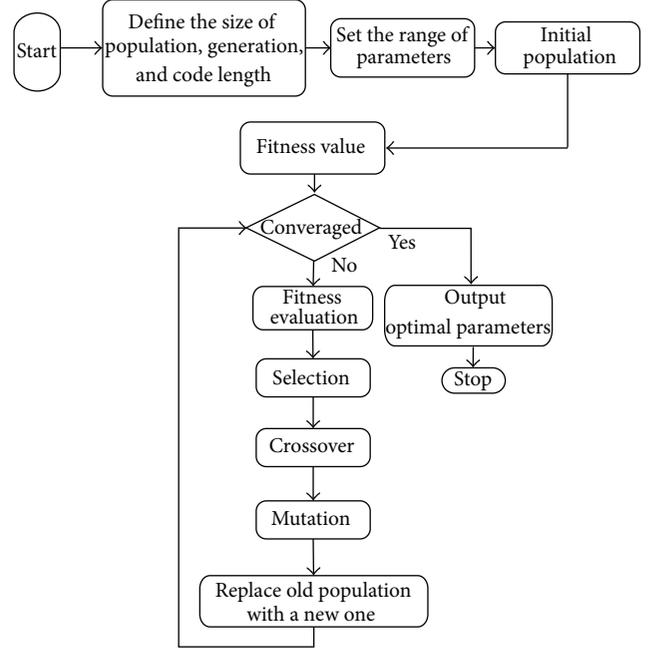


FIGURE 2: The process of GA optimization, when the convergence condition is satisfied, the optimal parameters will be obtained.

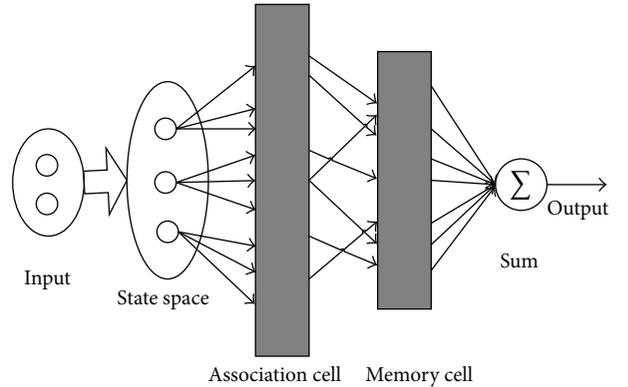


FIGURE 3: Structure of CMAC neural network.

3.4. CMAC Neural Network. The CMAC neural network has three steps: projecting an input into association area, compressing memory cell through Hash coding, and calculating the output as a scalar product of the memory area [41], which is shown in Figure 3. The output of CMAC can be expressed as follows [42]:

$$\begin{aligned} \mathbf{y}_s &= \mathbf{C}_s^T \mathbf{H} \mathbf{W} \\ &= [c_{s,1} \ c_{s,2} \ \cdots \ c_{s,N_h}] \begin{bmatrix} h_{1,1} & \cdots & h_{1,M_p} \\ \vdots & & \vdots \\ h_{N_h,1} & \cdots & h_{N_h,M_p} \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \vdots \\ \omega_{M_p} \end{bmatrix}, \quad (18) \end{aligned}$$

```

G = 200, Size = 30, CodeL = 10, parameters definition
input vector  $X_k = [q_{hd}(k), q_{kd}(k)]^T$ , the length of optimized parameters Len = 8
initialize population  $E = \text{round}(\text{rand}(\text{Size}, \text{Siez} * \text{CodeL}))$ 
for  $g = 1, 2, \dots, G$  do
  for  $s = 1, 2, \dots, \text{Size}$  do
    for  $l = 1, 2, \dots, \text{Len}$  do
       $F(s, l) = (\text{Max}(l) - \text{Min}(l)) * \text{Code}(j)/1023 + \text{Min}(l)$ ;  $F(s, l)$  will be used for fitness
    end for
  end for
  Selection and reproduction
  sort the fitness value and obtain the sequence number index
  for  $s = 1, 2, \dots, \text{Size}$  do
     $\text{TempE}(jj, :) = E(\text{index}(i), :)$ ;  $jj = jj + 1$ 
  end for
  Crossover and select the probability  $p_c = 0.8$ 
  for  $s = 1, 2, \dots, \text{Size}$  do
    temp = rand
    If  $p_c > \text{temp}$  do
      for  $t = 1, 1, \dots, \text{Num}$  do
         $\text{TempE}(s, t) = E(s + 1, t)$ 
         $\text{TempE}(s + 1, t) = E(s, t)$ 
      end for
    end if
  end for
  Mutation and select the probability
   $p_m = 0.001 - [1 : 1 : \text{Size}] * (0.001)/\text{Size}$ , temp = rand
  for  $s = 1, 2, \dots, \text{Size}$  do
    for  $j = 1, 2, \dots, \text{Len}$  do
      if  $p_m > \text{temp}$  do
        if  $\text{TempE}(s, j) == 0$  do
           $\text{TempE}(s, j) = 1$ 
        else
           $\text{TempE}(s, j) = 0$ 
        end if
      end if
    end for
  end for
  replace old generation with new one
end for
Obtain optimal parameters

```

ALGORITHM 1: Optimize SMC with GA. (Notation: $q_{hd}(k)$ and $q_{kd}(k)$ represent desired trajectory of hip joint and knee joint, resp.)

where \mathbf{C}_s is an association vector projected by input vector, \mathbf{W} is the weight vector, \mathbf{H} is the matrix of Hash coding, \mathbf{M}_p is the number of Hash vector, N_h is the number of association vector, and $h_{ij} = 1$ represents i th association unit response to j th Hash unit.

Similar to other neural networks, the weight parameters should be updated using Least Square Method (LSM). The updating process is expressed as follows:

$$\Delta \mathbf{W} = \frac{\eta}{N_h} \mathbf{A}_{s-1} (\hat{\mathbf{y}}_{s-1} - \mathbf{A}_{s-1}^T \mathbf{W}_{s-1}), \quad (19)$$

$$\mathbf{W}(k+1) = \mathbf{W}(k) + \Delta \mathbf{W} + \alpha (\mathbf{W}(k+1) - \mathbf{W}(k)),$$

where $\Delta \mathbf{W}$ is the weight vector increment, η is the learning rate, $\mathbf{A}_{s-1}^T = \mathbf{C}_{s-1}^T \mathbf{H}$, $\hat{\mathbf{y}}_{s-1}$ is the target output, and α is the inertial parameter.

CMAC was originally proposed to be applied into control problems by Miller III et al. [43]. The CMAC control loop is usually added to traditional control loops, where the traditional controller actuates the plant stably and the CMAC helps to improve control preciseness without affecting the traditional control loop [44, 45]. In other words, the CMAC control is usually added as a compensation item of traditional control method. The algorithm of the hybrid control strategy combining SMC and CMAC is shown as Algorithm 2 in Appendix B.

3.5. *Combination of GA Optimization-Based SMC and CMAC Neural Network.* Based on discussion above, we can combine SMC, GA, and CMAC neural network into a hybrid control strategy, which is called SMC.GA.CMAC.

```

input vector  $s = [s_1, s_2]$ , determine the range of  $s$  as  $s_{\max 1}, s_{\min 1}, s_{\max 2}, s_{\min 2}$ 
initialize CMAC, the range of quantization  $M$ , storage of association  $N$ , storage of memory  $C$ 
Input quantization:
 $s_1 = \text{round}((s_1 - s_{\min 1}) * M / (s_{\max 1} - s_{\min 1})), s_2 = \text{round}((s_2 - s_{\min 2}) * M / (s_{\max 2} - s_{\min 2}))$ 
Hash coding and obtain output of CMAC
for  $k = 1, 2, \dots, C$  do
     $\text{add}_1 = \text{mod}(s_1 + k, N) + 1, \text{add}_2 = \text{mod}(s_2 + k, N) + 1$ 
     $\text{Sum}_1 = w_1(\text{add}_1(k), \text{add}_2(k)), \text{Sum}_2 = w_2(\text{add}_2(k), \text{add}_2(k))$ 
end for
Weight update
for  $i = 1, 2, \dots, C$  do
    for  $j = 1, 2, \dots, C$  do
         $dw_1(i, j) = \eta e_1 / C, dw_2(i, j) = \eta e_2 / C$ 
    end for
end for
 $w_1 = w_{1,1} + dw_1 + \alpha(w_{1,1} - w_{1,2}), w_2 = w_{2,1} + dw_2 + \alpha(w_{2,1} - w_{2,2})$ 
 $w_{1,2} = w_{1,1}, w_{1,1} = w_1, w_{2,2} = w_{2,1}, w_{2,1} = w_2$ 

```

ALGORITHM 2: The process of CMAC neural network. (Notation: the input is sliding surface and the output is the compensation control vector.)

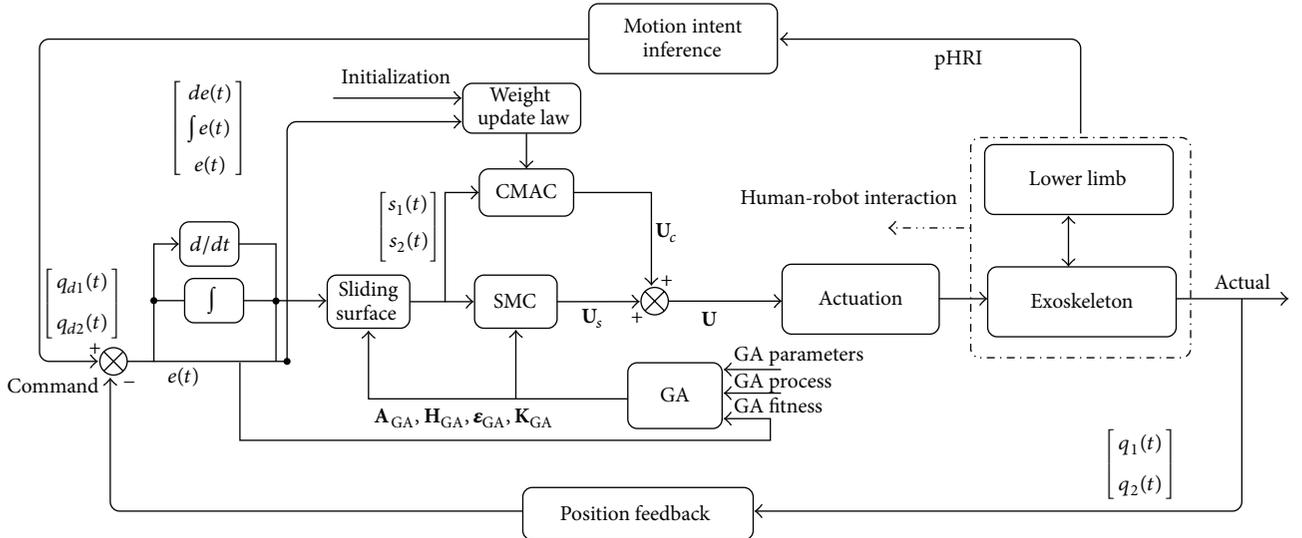


FIGURE 4: The control diagram of the proposed method for the exoskeleton system.

We can write the proposed control law based on (9) as follows:

$$\begin{aligned}
 \mathbf{u} = & \mathbf{K}_c \mathbf{U}_c + \mathbf{M}(q) \ddot{\mathbf{q}}_d - (\mathbf{T}_d - \mathbf{C}(q, \dot{q}) \dot{\mathbf{q}}) \\
 & + \mathbf{M}(q) \mathbf{A}_{GA} \dot{\mathbf{e}} + \mathbf{M}(q) \mathbf{H}_{GA} \mathbf{e} + \mathbf{C}(q, \dot{q}) \mathbf{s} \\
 & + \boldsymbol{\varepsilon}_{GA} \theta(\mathbf{s}) + \mathbf{K}_{GA} \mathbf{s},
 \end{aligned} \quad (20)$$

where \mathbf{U}_c represents the output of CMAC neural network, $\mathbf{K}_c \in \mathbb{R}^{n \times 1}$ is a positive definite matrix, and \mathbf{A}_{GA} , \mathbf{H}_{GA} , $\boldsymbol{\varepsilon}_{GA}$, and \mathbf{K}_{GA} are matrices optimized using GA. With the reaching condition (6), the output of CMAC \mathbf{U}_c has constraint as the following:

$$\mathbf{U}_c = \begin{cases} \mathbf{U}_c, & \text{if } \mathbf{s}^T \mathbf{U}_c \geq 0, \\ -\mathbf{U}_c, & \text{if } \mathbf{s}^T \mathbf{U}_c < 0. \end{cases} \quad (21)$$

For the exoskeleton system, the control diagram is illustrated as Figure 4 shows. As Figure 4 shows, GA is employed to obtain the optimal parameters \mathbf{A}_{GA} , \mathbf{K}_{GA} , \mathbf{H}_{GA} , and $\boldsymbol{\varepsilon}_{GA}$ to construct optimized SMC. The CMAC's input is the sliding surface and its weight updating is derived from minimizing the tracking error. The output of the proposed control law is $\mathbf{U} = \mathbf{U}_s + \mathbf{K}_c \mathbf{U}_c$, of which \mathbf{U}_s is the main output provided by SMC.GA and \mathbf{U}_c is the compensation output provided by CMAC.

4. Simulations with the Proposed Control Strategy

In this section, the proposed method is examined through simulations. The simulation results, which are from

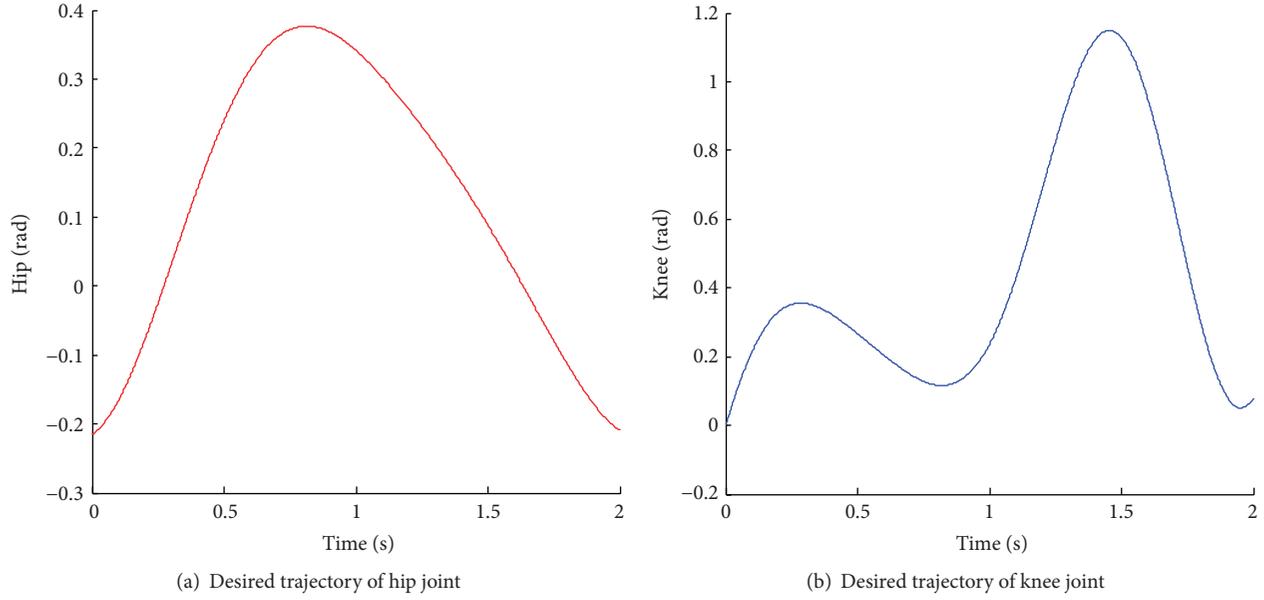


FIGURE 5: The desired joint trajectory of human lower limb movement. The initial posture is in the vertical direction.

application into controlling the swing leg of the exoskeleton using the proposed algorithms, are presented. As Figure 1 shows, the active DOFs are hip joint and knee joint, while the ankle joint is passive. Based on (1), the dynamics model of swing leg can be expressed as

$$\mathbf{M}(q)_{\text{exo}} \ddot{\mathbf{q}} + \mathbf{C}(q, \dot{q})_{\text{exo}} \dot{\mathbf{q}} + \mathbf{G}(q)_{\text{exo}} = \mathbf{T}, \quad (22)$$

where $\mathbf{M}(q)_{\text{exo}} \in \mathbb{R}^{2 \times 2}$, $\mathbf{C}(q, \dot{q})_{\text{exo}} \in \mathbb{R}^{2 \times 2}$, $\mathbf{G}(q)_{\text{exo}} \in \mathbb{R}^{2 \times 1}$, $\ddot{\mathbf{q}} \in \mathbb{R}^{2 \times 1}$, $\dot{\mathbf{q}} \in \mathbb{R}^{2 \times 1}$, and $\mathbf{T} \in \mathbb{R}^{2 \times 1}$. In simulations, the desired angular position of lower limb joints stems from the CGA data as Figure 5 shows. The period of the cyclical gait is 2 seconds and we will obtain the fitting expression with respect to time

$$\begin{aligned} q_{\text{hip}}(t) &= 3.85 \cos(0.330t + 2.14) \\ &\quad + 71.6 \cos(3.49t - 1.88) \\ &\quad + 41.0 \cos(4.68t - 0.3), \\ q_{\text{knee}}(t) &= 40.9 \cos(1.04t - 0.208) \\ &\quad + 157 \cos(5.82 - 0.047) \\ &\quad + 82.3 \cos(7.49t - 4.13), \end{aligned} \quad (23)$$

where $q_{\text{hip}}(t)$ and $q_{\text{knee}}(t)$ are the desired angular position of hip joint and knee joint, respectively.

To investigate the effectiveness and robustness of the proposed scheme, two simulation cases are considered: without disturbances (Case One) and with bounded disturbances (Case Two). The external disturbance $D(t)$ is a function of time which is assumed to have an upper bound:

$$D(t) = a \sin(\pi t), \quad \|a\| \leq 1. \quad (24)$$

For recording the respective performances, the root mean square error (RSME) is defined to examine control performance as follows:

$$\text{RSME} = \sqrt{\frac{\sum_{k=1}^N \|e(k)\|^2}{N}}, \quad (25)$$

where N is the size of error vector. We integrate ADAMS and MATLAB/SIMULINK to control the exoskeleton using the proposed control strategy, which is shown in Figure 6. As Figure 6 shows, there are six output variables from ADAMS model which are angular position, velocity, and acceleration of knee joint and hip joint of a swing leg while the designed controller in MATLAB outputs two control torques into ADAMS model. Figure 6(a) shows the exoskeleton model in ADAMS and Figure 6(b) shows the control scheme in SIMULINK. The designed controller produces control signals transferred to ADAMS while the kinematics information of exoskeleton joints is measured in ADAMS and returned back to MATLAB workspace. Through creating a communication block between MATLAB and ADAMS, the dynamics movements in gait cycles are shown in Figure 7, which illustrates the level ground walking for the lower extremity exoskeleton.

The comparisons between the proposed control scheme and conventional SMC, SMC with CMAC (SMC_CMAC) neural network, and optimal SMC with GA (SMC_GA) are conducted. The simulated comparisons, containing tracking positions and tracking errors of SMC, SMC_GA, SMC_CMAC, and SMC_GA_CMAC in Case One and Case Two, are depicted in Figures 8 and 9. As Figure 8 shows, Figures 8(a) and 8(c) represent the joint trajectory tracking of hip joint and knee joint while Figures 8(b) and 8(d) show tracking error comparisons of those two joints using

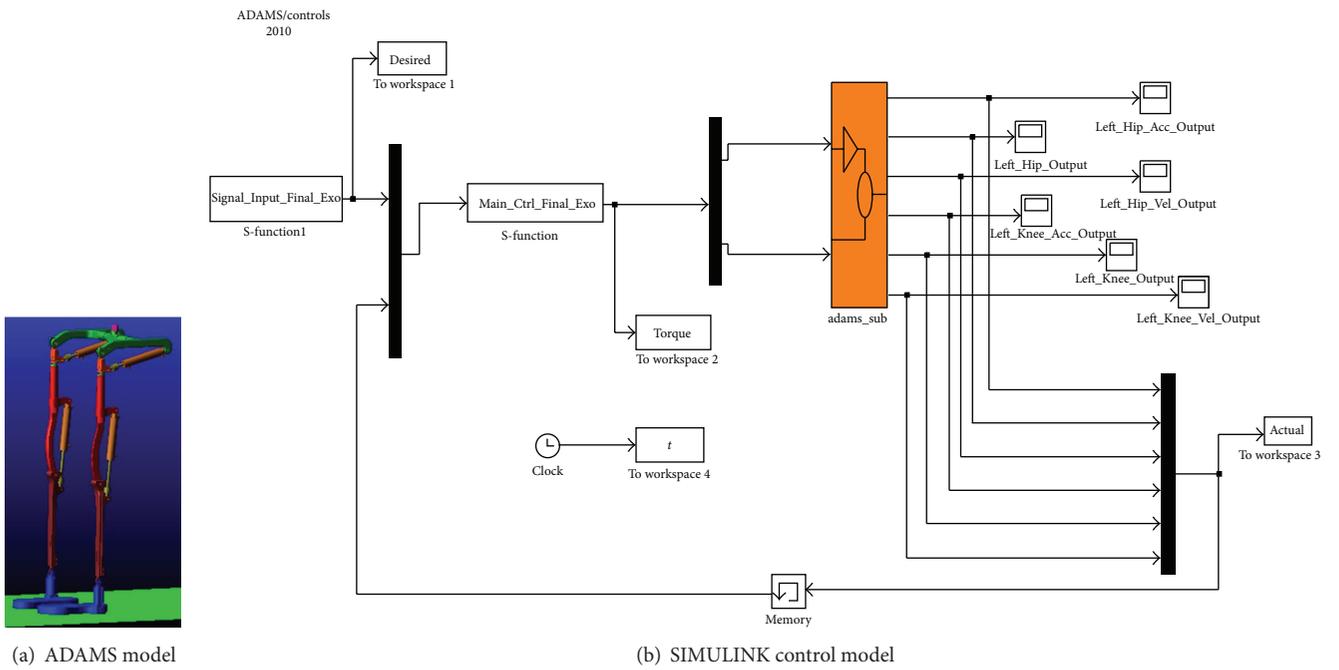


FIGURE 6: Cosimulation using ADAMS and MATLAB for exoskeleton robot.

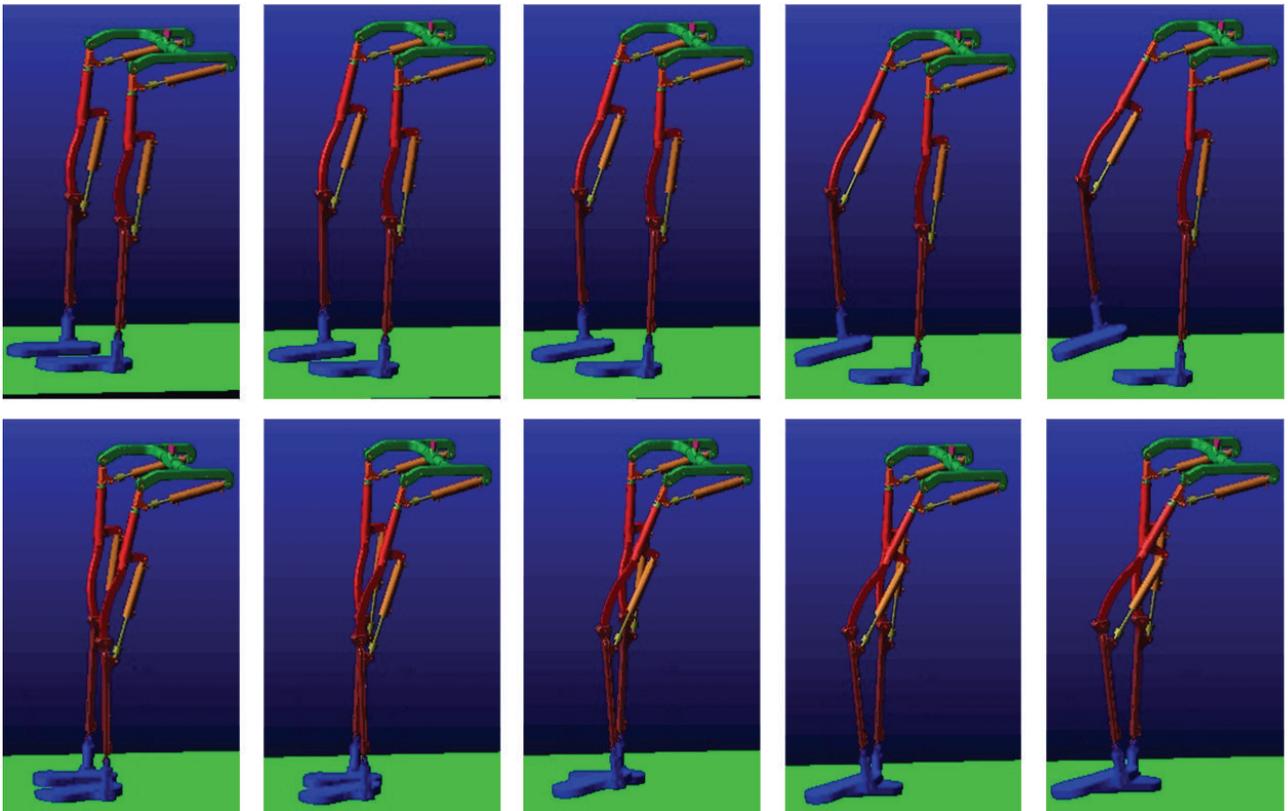


FIGURE 7: ADAMS effect pictures of gait cycles for lower extremity exoskeleton.

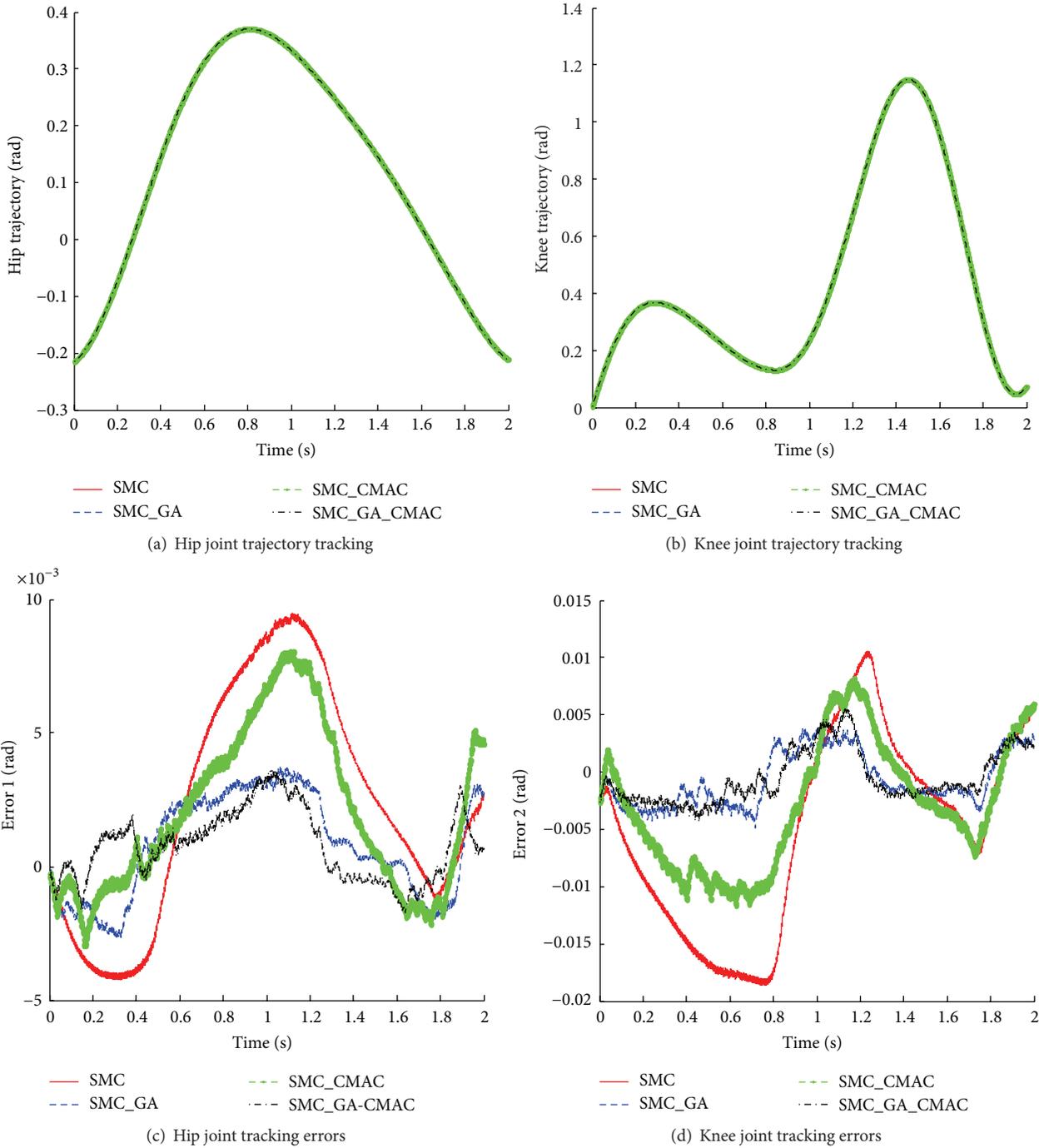


FIGURE 8: The performance comparisons using four methods in Case One.

four kinds of controllers separately. It can be seen that all of controllers can achieve good tracking performance and the conventional SMC without optimization has the largest tracking errors. Similarly, the angular position tracking and tracking errors comparisons in Case Two are depicted in Figures 9(a)–9(d). As Figure 9 shows, the desired joint angular trajectory also can be tracked well. To evaluate the control performances of Case One and Case Two, RSME comparisons using four controllers are depicted in

Figures 10 and 11. Figure 10(a) gives RSME of two joint tracking errors in Case One while Figure 10(b) describes that in Case Two. In two cases, the performance sequence from worse to better should be SMC, SMC_CMAC, SMC_GA, and SMC_GA_CMAC. Figure 11(a) illustrates the RSME comparison of hip joint while Figure 11(b) illustrates that of knee joint. In Figures 10 and 11, the RSME do not change much; hence the proposed control strategy still works when there exists external disturbance.

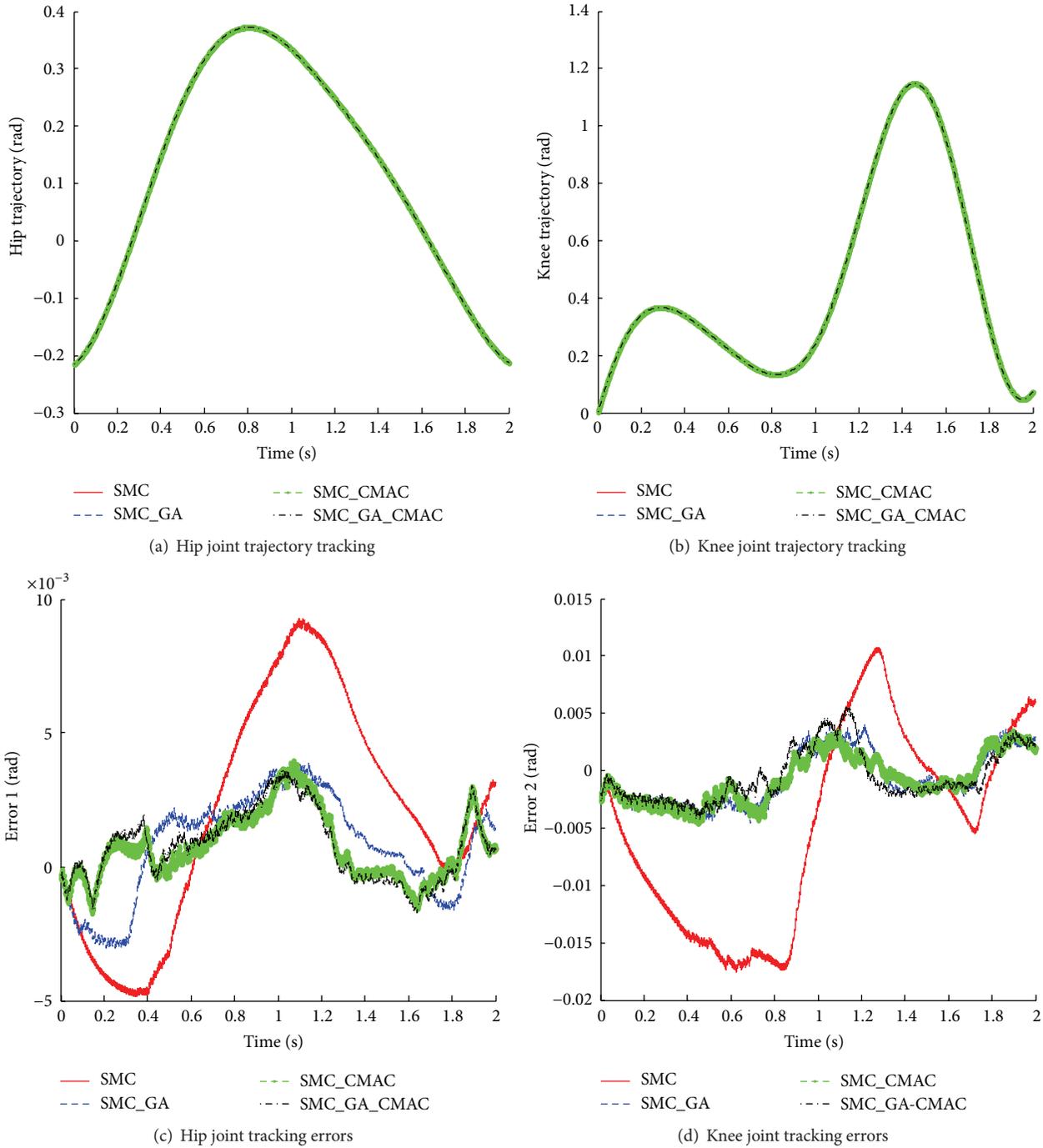


FIGURE 9: The performance comparisons using four methods in Case Two.

If the RSME is treated as a benchmark, the improvement percent (IMP) of performance with four kinds of controllers in Case One and Case Two is displayed in Tables 1 and 2. As the two tables show, the proposed control strategy will gain the highest improvement percentages of 69.4% and 76.8% for hip joint and knee joint separately in Case One while they change to be 68.1% and 76.8% in Case Two. Tables 1 and 2 illustrate that the SMC_GA is inferior to SMC_GA_CMAC but has better performance than SMC_CMAC, while

the SMC_CMAC is superior to SMC. Therefore, the proposed control strategy is robust and effective whether the exoskeleton system dynamics suffer from bounded external disturbance or not.

5. Conclusions

For lower limb assistive exoskeletons, precise position control is very important for the human-exoskeleton

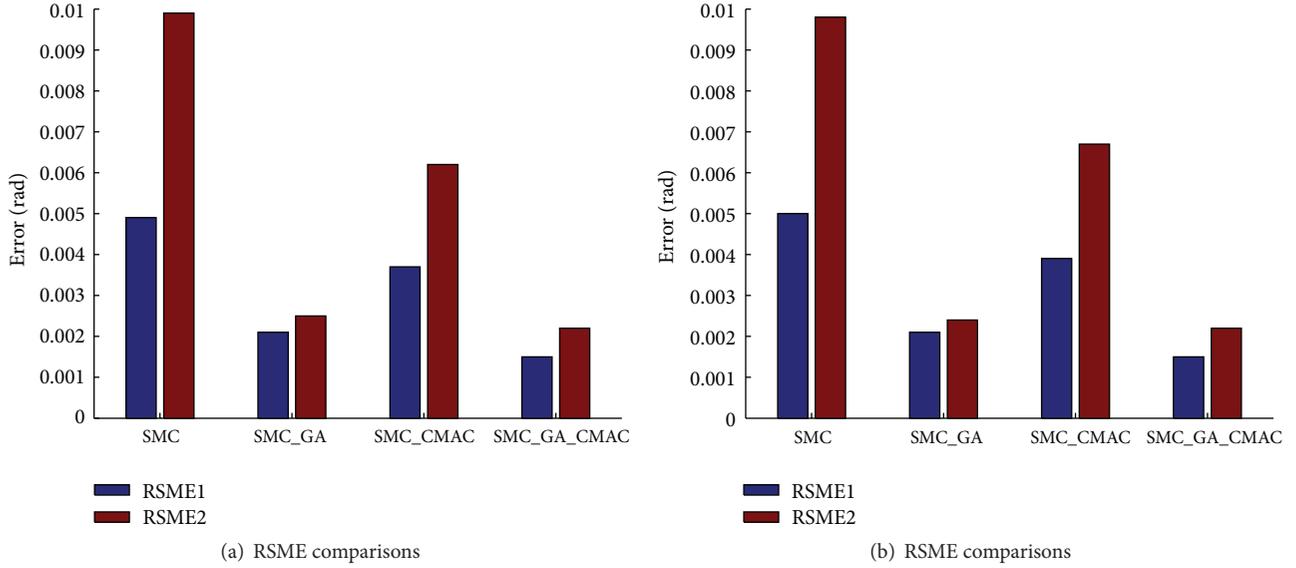


FIGURE 10: RSME comparisons in Case One (without disturbance) and Case Two (with bounded disturbance), respectively. RSME1 is for the hip joint while RSME2 is for the knee joint.

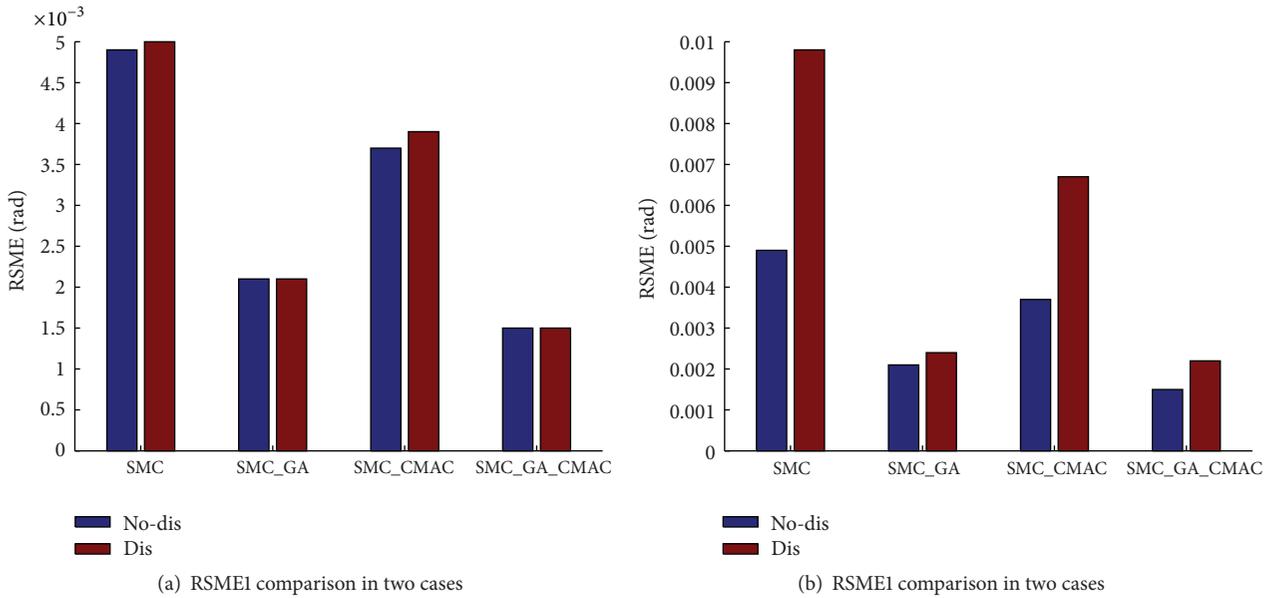


FIGURE 11: RSME1 and RSME2 comparisons in two cases. Icon “No-dis” means Case One and “Dis” means Case Two, respectively.

TABLE 1: Accuracy improvement comparison (Case One).

Control methods	IMP for hip joint (%)	IMP for knee joint (%)
SMC	0	0
SMC_GA	57.1%	74.5%
SMC_CMAC	24.5%	37.4%
SMC_GA_CMAC	69.4%	76.8%

TABLE 2: Accuracy improvement comparison (Case Two).

Control methods	IMP for hip joint (%)	IMP for knee joint (%)
SMC	0	0
SMC_GA	55.3%	75.2%
SMC_CMAC	17.4%	31.7%
SMC_GA_CMAC	68.1%	76.8%

collaboration. In this paper, a hybrid position control strategy SMC_GA_CMAC is proposed to follow human limb joints trajectory for the exoskeleton. GA is used to find

the optimal structure of SMC and CMAC neural network is implemented as the compensation to improve tracking performance. The proposed SMC_GA_CMAC control

strategy is proven to be stable with Lyapunov function and features better tracking performance compared with SMC, SMC_GA, and SMC_CMAC. The proposed control algorithm has guaranteed the requirement for high accuracy of position control for robotic manipulators suffering from dynamics uncertainties. The hybrid control strategy SMC_GA_CMAC is more suitable to control the exoskeleton to follow human motion intent under the occurrence of uncertainties. In addition, the proposed method will be investigated and explored in the real exoskeleton prototype in the near future.

Future study will be focused on the optimization of CMAC to overcome its drawbacks because of the binary input mapping character, which can be addressed by intelligent approaches such as fuzzy logic in cosimulation. The human motion intent estimation is also a crucial challenge, which will be investigated using machine learning methods.

Appendix

A. SMC Optimization Using GA

See Algorithm 1.

B. CMAC Neural Network Compensation

See Algorithm 2.

Competing Interests

The authors declare that they have no competing interests.

References

- [1] H. Kazerooni, R. Steger, and L. Huang, "Hybrid control of the Berkeley Lower Extremity Exoskeleton (BLEEX)," *International Journal of Robotics Research*, vol. 25, no. 5-6, pp. 561-573, 2006.
- [2] A. Zoss, H. Kazerooni, and A. Chu, "On the mechanical design of the Berkeley Lower Extremity Exoskeleton (BLEEX)," in *Proceedings of the IRS/RSJ International Conference on Intelligent Robots and Systems (IROS '05)*, pp. 3132-3139, August 2005.
- [3] <http://bleex.me.berkeley.edu/research/exoskeleton/hulc/>.
- [4] H. Kawamoto and Y. Sankai, "Power assist method based on Phase Sequence and muscle force condition for HAL," *Advanced Robotics*, vol. 19, no. 7, pp. 717-734, 2005.
- [5] A. Zelinsky, "Robot suit hybrid assistive limb," *IEEE Robotics and Automation Magazine*, vol. 16, no. 4, pp. 98-102, 2009.
- [6] S. Maeshima, A. Osawa, D. Nishio et al., "Efficacy of a hybrid assistive limb in post-stroke hemiplegic patients: a preliminary report," *BMC Neurology*, vol. 11, no. 1, article 116, 6 pages, 2011.
- [7] S. Yu, C. Han, and I. Cho, "Design considerations of a lower limb exoskeleton system to assist walking and load-carrying of infantry soldiers," *Applied Bionics & Biomechanics*, vol. 11, no. 3, pp. 119-134, 2014.
- [8] J. C. Moreno, F. Brunetti, E. Navarro, A. Forner-Cordero, and J. L. Pons, "Analysis of the human interaction with a wearable lower-limb exoskeleton," *Applied Bionics & Biomechanics*, vol. 6, no. 2, pp. 245-256, 2009.
- [9] M. H. Rahman, M. Saad, J.-P. Kenné, and P. S. Archambault, "Control of an exoskeleton robot arm with sliding mode exponential reaching law," *International Journal of Control, Automation and Systems*, vol. 11, no. 1, pp. 92-104, 2013.
- [10] X. T. Tran and H. J. Kang, "Adaptive hybrid high-order terminal sliding mode control of MIMO uncertain nonlinear systems and its application to robot manipulators," *International Journal of Precision Engineering and Manufacturing*, vol. 16, no. 2, pp. 255-266, 2015.
- [11] T. Hsiao and M.-C. Weng, "Robust joint position feedback control of robot manipulators," *Journal of Dynamic Systems, Measurement and Control*, vol. 135, no. 3, Article ID 031010, pp. 815-826, 2013.
- [12] H. G. Sage, M. F. de Mathelin, and E. Ostertag, "Robust control of robot manipulators: a survey," *International Journal of Control*, vol. 72, no. 16, pp. 1498-1522, 1999.
- [13] H. C. Cho, M. S. Fadali, K. S. Lee, and N. H. Kim, "Adaptive position and trajectory control of autonomous mobile robot systems with random friction," *IET Control Theory & Applications*, vol. 4, no. 12, pp. 2733-2742, 2010.
- [14] S. A. M. Dehghan, M. Danesh, and F. Sheikholeslam, "Adaptive hybrid force/position control of robot manipulators using an adaptive force estimator in the presence of parametric uncertainty," *Advanced Robotics*, vol. 29, no. 4, pp. 209-223, 2015.
- [15] R.-J. Wai, Y.-C. Huang, Z.-W. Yang, and C.-Y. Shih, "Adaptive fuzzy-neural-network velocity sensorless control for robot manipulator position tracking," *IET Control Theory & Applications*, vol. 4, no. 6, pp. 1079-1093, 2010.
- [16] J. Y. Cheong, S. I. Han, and J. M. Lee, "Adaptive fuzzy dynamic surface sliding mode position control for a robot manipulator with friction and deadzone," *Mathematical Problems in Engineering*, vol. 2013, Article ID 161325, 15 pages, 2013.
- [17] V. Utkin, J. Guldner, and J. Shi, *Sliding Mode Control in Electromechanical Systems*, CRC Press, New York, NY, USA, 2009.
- [18] F. G. Rossomando, C. Soria, and R. Carelli, "Sliding mode neuro adaptive control in trajectory tracking for mobile robots," *Journal of Intelligent & Robotic Systems*, vol. 74, no. 3-4, pp. 931-944, 2014.
- [19] M. Taherkhorsandi, M. J. Mahmoodabadi, M. Talebipour, and K. K. Castillo-Villar, "Pareto design of an adaptive robust hybrid of PID and sliding control for a biped robot via genetic algorithm optimization," *Nonlinear Dynamics*, vol. 79, no. 1, pp. 251-263, 2014.
- [20] B. S. K. K. Ibrahim, R. Ngadengon, and M. N. Ahmad, "Genetic algorithm optimized integral sliding mode control of a direct drive robot arm," in *Proceedings of the International Conference on Control, Automation and Information Sciences (ICCAIS '12)*, pp. 328-333, Ho Chi Minh City, Vietnam, November 2012.
- [21] M. R. Soltanpour and M. H. Khooban, "A particle swarm optimization approach for fuzzy sliding mode control for tracking the robot manipulator," *Nonlinear Dynamics*, vol. 74, no. 1-2, pp. 467-478, 2013.
- [22] Y.-H. Chang, C.-W. Chang, C.-W. Tao, H.-W. Lin, and J.-S. Taur, "Fuzzy sliding-mode control for ball and beam system with fuzzy ant colony optimization," *Expert Systems with Applications*, vol. 39, no. 3, pp. 3624-3633, 2012.
- [23] C. Su, G. Lii, and H. Hwung, "Position control employing fuzzy-sliding mode and genetic algorithms with a modified evolutionary direction operator," *International Journal of Cybernetics & Systems*, vol. 30, pp. 873-891, 2010.
- [24] E. Fuchs and M. A. S. Masoum, *Power Quality in Power Systems and Electrical Machines*, Academic Press, New York, NY, USA, 2011.

- [25] S. J. Got, M. C. Lee, and M. K. Park, "Fuzzy-sliding mode control of a polishing robot based on genetic algorithm," *Journal of Mechanical Science & Technology*, vol. 15, pp. 580–591, 2001.
- [26] M. J. Kharaajoo and H. Rouhani, *Advances in Artificial Intelligence*, Springer, Berlin, Germany, 2004.
- [27] M. J. Mahmoodabadi, M. Taherkhorsandi, M. Talebipour, and K. K. Castillo-Villar, "Adaptive robust PID control subject to supervisory decoupled sliding mode control based upon genetic algorithm optimization," *Transactions of the Institute of Measurement & Control*, vol. 37, no. 4, pp. 505–514, 2015.
- [28] A. R. Firdaus and A. S. Rahman, "Genetic algorithm of sliding mode control design for manipulator robot," *Telecommunication Computing Electronics and Control*, vol. 10, pp. 645–654, 2012.
- [29] J. S. Albus, "A new approach to manipulator control: the cerebellar model articulation controller (CMAC)," *Journal of Dynamic Systems, Measurement, and Control*, vol. 97, pp. 220–227, 1975.
- [30] W. Yu, M. A. Moreno-Armendariz, and F. O. Rodriguez, "Stable adaptive compensation with fuzzy CMAC for an overhead crane," *Information Sciences*, vol. 181, no. 21, pp. 4895–4907, 2011.
- [31] H. Duan and D. Gu, "Sliding mode adaptive control for flying robot based on recurrent CMAC algorithm," in *Proceedings of the IEEE International Conference on Mechatronics and Automation (ICMA '11)*, pp. 440–445, Beijing, China, August 2011.
- [32] A. Chu, *Design of the Berkley lower extremity exoskeleton (BLEEX) [Ph.D. thesis]*, University of California, Berkeley, Calif, USA, 2005.
- [33] C. Kirtley, CGA Normative Gait Database, <http://www.clinical-gaitanalysis.com/data/>.
- [34] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*, John Wiley & Sons, New York, NY, USA, 2006.
- [35] J.-H. Lee, "Highly robust position control of BLDDSM using an improved integral variable structure systems," *Automatica*, vol. 42, no. 6, pp. 929–935, 2006.
- [36] J. J. Slotine and W. Li, *Applied Nonlinear Control*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1991.
- [37] Y. Long and X.-J. Yang, "Robust adaptive fuzzy sliding mode synchronous control for a planar redundantly actuated parallel manipulator," in *Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO '12)*, pp. 2264–2269, Guangzhou, China, December 2012.
- [38] M. Krstic, I. Kanellakopoulos, and P. V. Kokotovic, *Nonlinear and Adaptive Control Design*, John Wiley & Sons, New York, NY, USA, 1995.
- [39] E. Köse, K. Abacı, H. Kızmaz, S. Aksoy, and M. A. Yalçın, "Sliding mode control based on genetic algorithm for WSCC systems include of SVC," *Electronics and Electrical Engineering*, vol. 19, no. 4, pp. 25–28, 2013.
- [40] M. Mitchell, *An Introduction to Genetic Algorithms*, MIT Press, Cambridge, Mass, USA, 5th edition, 1999.
- [41] G. Horváth, "Kernel CMAC: an efficient neural network for classification and regression," *Acta Polytechnica Hungarica*, vol. 3, no. 1, pp. 5–20, 2006.
- [42] X. J. Yang and Y. Long, "Synchronous trajectory tracking control and simulation of CMAC neural network based on computed torque control," *Journal of Harbin Institute of Technology*, vol. 45, no. 7, pp. 85–89, 2013.
- [43] W. T. Miller III, R. P. Hewes, F. H. Glanz, and L. G. Kraft, "Real-time dynamic control of an industrial manipulator using a neural network-based learning controller," *IEEE Transactions on Robotics and Automation*, vol. 6, no. 1, pp. 1–9, 1990.
- [44] F.-C. Chen and C.-H. Chang, "Practical stability issues in CMAC neural network control systems," *IEEE Transactions on Control Systems Technology*, vol. 4, no. 1, pp. 86–91, 1996.
- [45] K. Mohajeri, G. Pishehvar, and M. Seifi, "CMAC neural networks structures," in *Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA '09)*, pp. 39–45, Daejeon, Republic of Korea, December 2009.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

