

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800

Open access books available

122,000

International authors and editors

135M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



A CMAC-Based Systematic Design Approach of an Adaptive Embedded Control Force Loading System

Jian Chen, Peng Li, Gangbing Song, Shubo Wang,
Zichao Zhang, Guangqi Wang, Yu Tan and
Yongjun Zheng

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/intechopen.71420>

Abstract

In this chapter, an adaptive embedded control system is developed to measure yield strength of the material plate with an applied load. A systematic approach is proposed to handle special requirements of embedded control systems which are different from computer-based control systems as there are much less computational power and hardware resources available. Efficient control algorithm has to be designed to remove CPU burden so that the microcontroller has enough power available. A three-step approach is proposed to address the embedded control issue: Firstly, the mathematical description of the whole system is studied using both theoretical and experimental methods. A mathematical model is derived from the physical models of each component used, and an experiment is retrieved by employing Levy's method and least square estimation to identify specific parameters of the system model. Secondly, an adaptive feedforward plus feedback controller is designed and simulated as a preparation for the embedded system implementation. The Cerebellar Model Articulation Controller (CMAC) is chosen as the feedforward part, and a PD controller is used as the feedback part to train the CMAC. Finally, the proposed algorithm is applied to the embedded system, and experiments are conducted to verify both the identified model and designed controller.

Keywords: embedded system, force control, systems modeling, systems identification, cerebellar model articulation controller

1. Introduction

With the increasing threat of global warming and the waning of fossil fuel reserves, renewable energy is playing more and more important roles in both environmental and economic aspect

of lives. Solar photovoltaic (PV), as one of the most important renewable energy sources, has been widely used in more than a hundred of countries [1]. Solar panels are commonly deployed in harsh environments with high temperature, high humidity and strong winds [2], and are expected to be functional for over 30 years. Thus the glass panel, where the PV material will be located, has to be specially designed to sustain the harsh environments over its life of service [3]. As a necessary validation step, the strength and quality of the manufactured glass sheet must be tested and qualified, especially before the mass production stage. A rigorous test would require the strength of the solar glass sheet to be verified under a controlled environment with simulated temperature, humidity and induced vibration or forces on the PV glass sheet. Although commercial solutions exist for the various types of material-strength testing, environment simulated glass sheet strength testing is not available. As the booming of the PV industry and the increasing researching into stronger and better PV glass sheet, there has been a greater demand into the research and development of specially designed PV glass sheet test machines. This chapter presented the design of an environment controlled force loading machine specifically for solar panel glass sheet strength testing.

The glass sheet testing machine is basically a real-time electro-mechanical control system which is composed of actuators for simulated loading generation, sensors for monitoring and control feedback, a digital controller for the control of the overall system performance and mechanical supporting structures to hold everything together. The key part is the real-time digital controller which serves as the brain of such a system. Although computer-based control system with great computational power is able to host complex control algorithm, testing facilities with space, power and budget constraint cannot afford the size, power and cost of the computer-based design. Moreover, many extended and long hour testing scenarios require great robustness on the control system, where computer with standard operating system cannot meet the requirement. Embedded control systems with much lower power consumption, compact space, less cost and increased robustness is preferred over computer-based systems in many custom designed solutions [4–6]. Although embedded control systems have many advantages, they usually have limited or reduced computational powers that may not be able to host complex control algorithms. This poses challenges into the design of efficient real-time control algorithms. Specific embedded system-oriented control techniques must be carried out to guarantee control performances. For this reason, a simple and effective control law is preferred on the embedded control system and the Cerebellar Model Articulation Controller (CMAC) is a good choice.

The CMAC proposed by Albus [7] is a lookup-table adaptive neural network for estimating complicated nonlinear functions. The basic idea of CMAC is to quantify states from input, find memory addresses of states according to their locations in memory, add the content in the memory address, generate the CMAC output, compare the output with desired output and update the content in memory based on learning algorithms. Compared with other neural networks, advantages of the CMAC [8] are: (1) it is based on local learning and stores information in local memory, hence weights are changed slightly in each step and its learning speed is fast and suitable for real-time control; (2) it owns definite generalization property, so that the close inputs generate the close outputs, and the different inputs produce the

different outputs; (3) it can receive continuous input and produce continuous output; (4) it can accelerate the response speed by using addressing mode and (5) it is a nonlinear approximation and robust to the sequences of training data. Owing to above predominant merits, the nonlinear approximating capacity of the CMAC is superior to that of other traditional neural networks and is more suitable for real-time control in the real world.

In summary, various CMAC control algorithms have been formed so far, such as CMAC feedforward control [9–11], CMAC feedback control [12–14], CMAC optimal control [15, 16], CMAC fuzzy control [17–22], CMAC H-infinity control [23] and CMAC adaptive control [24–26]. The previous works can be divided into two categories: one is to improve the control structures of the CMAC, such as the CMAC feedforward control; the other is to improve the learning algorithms with other intelligent techniques, such as the fuzzy CMAC (FCMAC). Although a lot of complicated advanced CMAC control algorithms can well perform the control tasks in the simulations or with the computer-based control systems, the performances raised by some of them are very limited compared with the basic CMAC structure, even several advanced CMAC control algorithms, such as the FCMAC with the wavelet, cannot be implemented in real time with an embedded system due to the complicated improvement algorithms.

In this chapter, we adopt CMAC feedforward and PD feedback control, because [10, 11]: (1) the CMAC carries out the feedforward part to approximate the inverse model of the plant; (2) the PD controller actualizes the feedback part to train the CMAC and guarantee the stability of the closed-loop system; (3) compared with other neural networks or other complicated CMACs, limited computation cost of the simple and effective CMAC plus PD control removes CPU burden, so that the microcontroller can have enough computational power available.

Since there is not much effort dealing with the systematic design problem of an embedded-control glass strength testing machine, the contribution of this chapter is to present a three-step systematic design approach to address the embedded control issue: Firstly, the mathematical description of the system is studied using both theoretical and experimental method. A mathematical model is derived from the physical models of each component used, and an experiment is retrieved by employing Levy's method to identify parameters of the mathematical model. Secondly, an adaptive CMAC feedforward plus PD feedback controller is designed and simulated based on the identified system model as a preparation for the embedded system implementation. Finally, the proposed algorithm is applied to the embedded system with the same parameters as those of simulations, and experiments are conducted to verify both the identified model and designed controller. To design a machine systematically for practical use, these three steps are closely linked and indispensable. The three-step systematic design approach could benefit engineers in measurement and control as a guide.

Rest of this chapter is organized as follows. Section 2 analyses the system requirements and formulates a mathematical system model. Section 3 performs the system identification. Section 4 proposes the simulation-based controller design. Embedded system-based experiments are conducted in Section 5 and Section 6 concludes this work.

2. System Modeling

2.1. System analysis

The schematic diagram of the force loader unit is shown in **Figures 1** and **2**. The loader unit consists of one loader and one steel bar, which is connected to the load cell and linear motor actuator. The linear motor actuator, used to apply force, consists of a motor with linear motion and an encoder. The loader is attached to the steel bar and then through a load cell to the linear motor. In short, the force is applied on the material plate by a loader unit which is connected to a linear motor actuator through a steel bar, and the applied force is recorded by a force sensor.

The load applied on the material plate varies as a ramp function. The user can select the slope of the ramp function by setting the maximum force in a finite time period on the touch screen, and can perform the test under different forces ranging from 0 to 300 lbs. The control objective is to ensure the applied force track the reference force command for measuring the material plate's yield stress qualified or not.

2.2. Mathematic model of the motor-loader unit

DC motors are widely used as actuators for high-precision servo control owing to its good working characteristics and simple mathematical model. Mechanical resonance phenomena are ubiquitous because the transmission shaft is not completely rigid and will be distorted under force. For the servo-motor-drive system, considering the mechanical resonance phenomena, the double-mass structure model is commonly used to describe such dynamical systems.

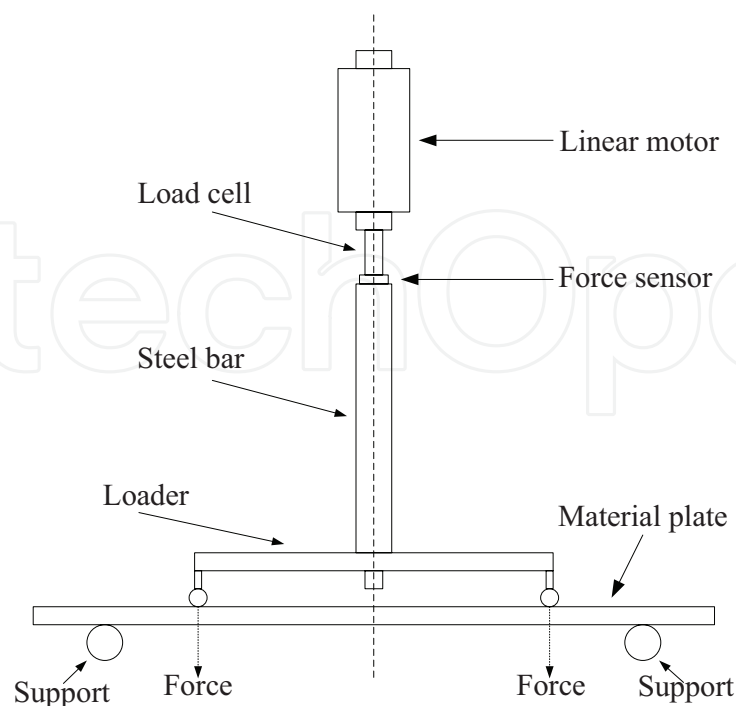


Figure 1. Schematic drawing of the loader unit.

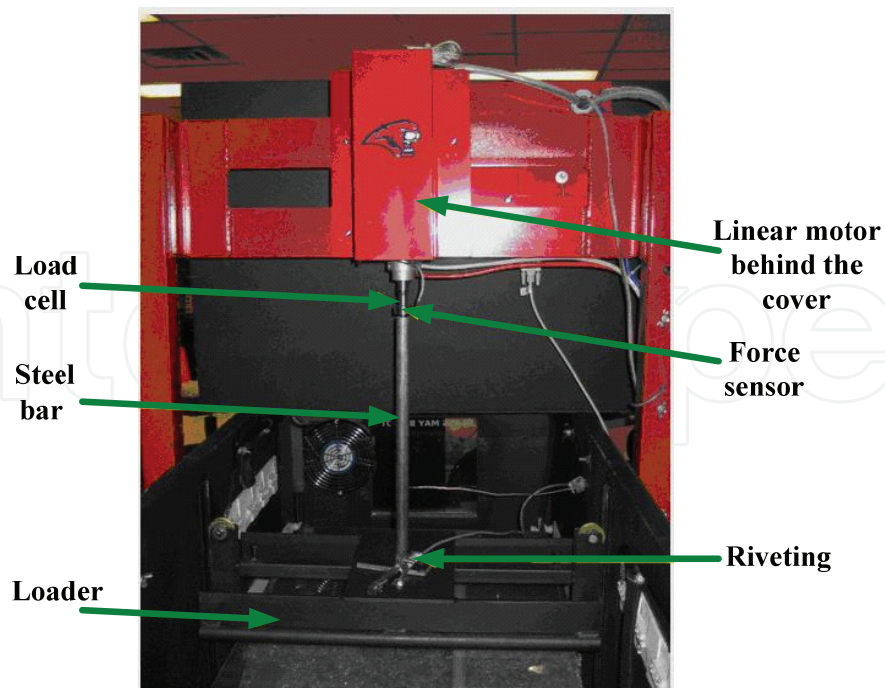


Figure 2. Photo of the loader unit.

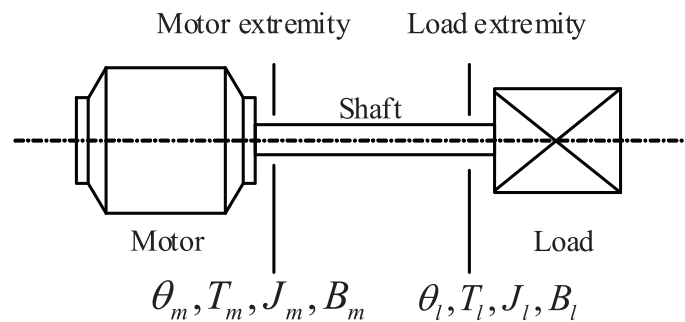


Figure 3. Motor and load double mass diagram.

The electrical equilibrium equation can be written as [27]

$$R_a i + L_a \frac{di}{dt} + K_e \dot{\theta}_m = u_m \quad (1)$$

where R_a is the armature resistance, L_a is the armature inductance, i is the armature current, K_e is the counter-electromotive force coefficient, θ_m is the motor rotating angle and u_m is the motor voltage. $K_e \dot{\theta}_m$ reflects that the back electromotive force (EMF) has a linear relationship with the motor speed. **Figure 3** illustrates the mechanical parameter of the motor with a load.

The motor output torque T_m is determined by the armature current as

$$T_m = K_t i \quad (2)$$

where K_t is the electromagnetism-torque constant.

The torque equilibrium equation of the motor is

$$J_m \ddot{\theta}_m + B_m \dot{\theta}_m = T_m - T_l \quad (3)$$

where J_m is the moment of inertia of the motor, B_m is the viscosity damping coefficient of the motor extremity and T_l is the torque of the load extremity. T_l can be represented as

$$T_l = K_s(\theta_m - \theta_l) \quad (4)$$

where θ_l is the rotating angle of the load shaft and K_s is the mechanical rigidity of the rotating shaft.

The torque equilibrium equation of the load is

$$J_l \ddot{\theta}_l + B_l \dot{\theta}_l = T_l - T_d \quad (5)$$

where J_l is the moment of inertia of the load, B_l is the damping coefficient on the load side and T_d is the disturbance torque, including the friction torque, the coupling torque and the external disturbance torque. The displacement of the loader mass generated from the linear motor has a linear relationship with the rotating angle of the load shaft, which can be represented as

$$Dis = K_D \cdot \theta_l \quad (6)$$

The force generated on the solar panel glass has an almost linear relationship with the deformation of the glass, which can be represented as

$$F_m = K_F \cdot Dis \quad (7)$$

2.3. Mathematical models of other components

The control objective is to track the input command through feedback control based on the signal measured by the force sensor. In addition to the motor-loader and the micro controller unit (MCU), other components included in the control loop are a digital to analog converter (DAC), an amplifier and a load cell. **Figure 4** illustrates the open-loop plant structure.

The DAC converter is basically a zero-order hold. Assume the controller's output is $u(t)$, the amplifier's input is $u_h(t)$ and the system's sampling time is T . The zero-order hold can be represented as

$$u_h(t) = u(kT), kT \leq t < (k+1)T \quad (8)$$

Considering the sampling process, by replacing e^{-Ts} with $e^{-Ts} \approx (-s + \frac{2}{T}) / (s + \frac{2}{T})$, the DAC converter and the zero-order hold can be described as

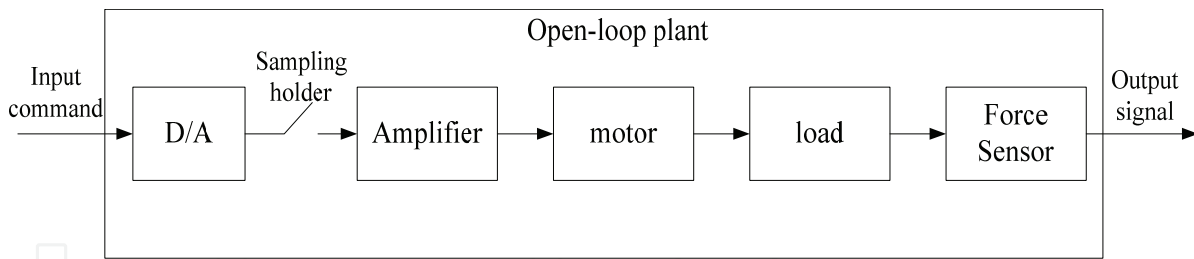


Figure 4. Open-loop block diagram.

$$G_H(s) = \frac{u_n(s)}{u(s)} \cdot \frac{1}{T} = \frac{1 - e^{-Ts}}{s} \cdot \frac{1}{T} \approx \frac{T}{\frac{T}{2}s + 1} \cdot \frac{1}{T} \quad (9)$$

which is a low-pass filter.

In a high-precision servo system, PWM-based amplifiers are commonly used as the motor drivers [28]. The PWM power amplifiers can be represented as

$$G_A(s) = \frac{K_a}{1 + \frac{1}{2f_s}s} \quad (10)$$

where K_a is the amplifying multiple and f_s is the modulation frequency.

The load cell signal conditioner linearly converts the force into a voltage signal, and its model can be simplified as

$$G_S(s) = K_{Sen} \quad (11)$$

2.4. Simplified model

Without considering the disturbance torque T_d , the angles of the motor and the load have the following dynamical relationship

$$T_l = J_l \ddot{\theta}_l + B_l \dot{\theta}_l = K_s(\theta_m - \theta_l) \Rightarrow \frac{\theta_l(s)}{\theta_m(s)} = \frac{K_s}{J_l s^2 + B_l s + K_s} \quad (12)$$

K_s is very large and the time constant is very small, therefore in the lower frequency range $\theta_l(s)/\theta_m(s) \approx 1$, $\theta_l \approx \theta_m$.

By considering $K_s = \infty$ and $\theta_l \equiv \theta_m$, the motor-loader model can be simplified as

$$J_a \ddot{\theta}_m + B_a \dot{\theta}_m = T_m \quad (13)$$

where $J_a = J_m + J_l$, $B_a = B_m + B_l$. Deleting the intermediate variables, the dynamical equation of the motor can be obtained as

$$\frac{L_a J_a}{K_t} \frac{d^3 \theta_m}{dt^3} + \frac{R_a J_a + L_a B_a}{K_t} \frac{d^2 \theta_m}{dt^2} + \left(K_e + \frac{R_a B_a}{K_t} \right) \frac{d \theta_m}{dt} = u_m \quad (14)$$

If the motor armature inductance is regarded as $L_a \approx 0$, the system can be further simplified as

$$\frac{R_a J_a}{K_t} \frac{d^2 \theta_m}{dt^2} + \left(K_e + \frac{R_a B_a}{K_t} \right) \frac{d \theta_m}{dt} = u_m \quad (15)$$

And the transfer function between the input voltage and output rotational angle is

$$G_M(s) = \frac{\theta_m(s)}{u_m(s)} = \frac{K_m}{T_m s^2 + s} \quad (16)$$

where $T_m = \frac{R_a J_a}{K_e K_t + R_a B_a}$ is the time constant of the motor system, and $K_m = \frac{K_t}{K_e K_t + R_a B_a}$ is the gain coefficient of the motor system.

2.5. Open-loop transfer function

The mathematical model of the open-loop plant is the cascade of the units described previously:

$$G_p(s) = G_H(s) \cdot G_A(s) \cdot G_M(s) \cdot G_S(s) \quad (17)$$

Because the system bandwidth is much lower than the system sampling rate and the modulation frequency of the PWM amplifier, the time constant of the DAC converter and the amplifier is very small. In the lower frequency range, the effect of the DAC converter and the amplifier can be omitted. Therefore, the open-loop plant can be simplified as

$$G_p(s) = \frac{b_1 s + b_0}{a_2 s^2 + a_1 s + 1} \quad (18)$$

where b_0, b_1, a_1 and a_2 are final parameters.

Obvious, it is a second-order model. Only four parameters need to be identified.

3. Parameters identification of the system model

Levy's method [29] and least square estimation are widely used in system identification. Assume that the identified transfer function is given as

$$G_p(s) = \frac{b_0 + b_1 s + \dots + b_m s^m}{1 + a_1 s + \dots + a_n s^n} \quad (19)$$

where $b_0, b_1, \dots, b_m, a_1, a_2, \dots, a_n$ are real numbers, m and n are integers, and $m \leq n$.

The frequency response is

$$G_p(j\omega) = \frac{(b_0 - b_2 \omega^2 + \dots) + j\omega(b_1 - b_3 \omega^2 + \dots)}{(1 - a_2 \omega^2 + \dots) + j\omega(a_1 - a_3 \omega^2 + \dots)} = \frac{N(j\omega)}{D(j\omega)} \quad (20)$$

For each frequency point $\omega_i (i = 1, 2, \dots, L)$, it is assumed that the actual frequency response is $\text{Re}(\omega_i) + j \text{Im}(\omega_i)$, and the approximation error is defined as

$$\varepsilon(j\omega_i) = \text{Re}(\omega_i) + j \text{Im}(\omega_i) - \frac{N(j\omega_i)}{D(j\omega_i)} \quad (21)$$

Define an objective function:

$$J = \sum_{i=1}^L \|D(j \omega_i) \varepsilon(j \omega_i)\|^2 \tag{22}$$

Minimize the objective function, and prompt $\partial J/\partial a_j = 0$ ($j = 1, 2, \dots, n$) and $\partial J/\partial b_k = 0$ ($k = 1, 2, \dots, m$), then two matrix equations will be obtained. By solving the matrix equations, we can get the estimated parameters $b_0, b_1, \dots, b_m, a_1, a_2, \dots, a_n$.

Table 1 shows actual frequency response obtained by experiments. We employ different sinusoidal input $r(t) = A_m \sin \omega t$ with different angular frequency ω (from 0.1 to 1 rad/s) to excite the open-loop system. By theoretical analysis, the output signals are in the form $y(t) = A_f \sin(\omega t + \Phi)$. Comparing the output with input sinusoidal signals, we can calculate the values of A_f/A_m and phase delay angle Φ based on the least square estimation as follows.

The output signals can be decomposed as

$$\begin{aligned} y(t) &= A_f \sin(\omega t + \Phi) = A_f \cos \Phi \sin \omega t + A_f \sin \Phi \cos \omega t \\ &= [\sin \omega t \quad \cos \omega t] \begin{bmatrix} A_f \cos \Phi \\ A_f \sin \Phi \end{bmatrix} \end{aligned} \tag{23}$$

First, we select the sampling interval $t = 0, h, 2h, \dots, nh$, where h is a step time.

Second, by defining $Y = [y(0) \ y(h) \ \dots \ y(nh)]^T$, $c_1 = A_f \cos \Phi$, $c_2 = A_f \sin \Phi$ and $\Psi = \begin{bmatrix} \sin(\omega 0) & \sin(\omega h) & \dots & \sin(\omega nh) \\ \cos(\omega 0) & \cos(\omega h) & \dots & \cos(\omega nh) \end{bmatrix}$, the least square solutions of c_1 and c_2 can be derived as

$$\begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = (\Psi^T \Psi)^{-1} \Psi^T Y \tag{24}$$

Third, A_f/A_m and Φ are calculated as

$$\frac{\hat{A}_f}{A_m} = \frac{\sqrt{c_1^2 + c_2^2}}{A_m}, \quad \Phi = \text{tg}^{-1}(c_2/c_1) \tag{25}$$

where \hat{A}_f is the estimation of A_f .

After A_f/A_m and Φ are obtained, it is easy to estimate the parameters b_0, b_1, a_1 and a_2 in Eq. (18) based on Levy's method. Finally, we get the identified transfer function:

$$G(s) = \frac{-38.48s - 141.4}{1.523 s^2 + 8.324s + 1} \tag{26}$$

ω	1	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1	1
A_f/A_m	16	22	25	27	30	31	40	46	64	122	16
Φ	1.71	1.82	1.78	1.77	1.89	1.92	1.90	2.00	1.94	2.23	1.71

Table 1. Actual frequency response (ω is angular frequency, rad/s; Φ is phase delay, rad).

4. Simulation-based controller design

4.1. Control scheme

As shown in **Figure 5**, in the CMAC plus PD control scheme, the PD controller is used to train the weights of the CMAC at the early stage of control. The output u_n of the CMAC and the output u_p of the PD controller are integrated as the control command u to track the desired input command. Once the system is running, the PD controller will play a main role at the beginning. As the weights are tuned by u_p , u_n will increase to become the main control command. The fast learning speed prompts the leading role of control to switch from the PD controller to the CMAC.

CMAC employs the supervisory learning algorithm. At the end of each control step k , the controller calculates the corresponding output $u_n(k)$, compares it with total control output $u(k)$, amends the weights and enters into learning process to make error small between the total control output and the output of the CMAC. That means the total control output will be finally generated by the CMAC only via its learning strategy.

The control law can be described as

$$\begin{cases} u(k) = u_n(k) + u_p(k) \\ u_n(k) = \sum_{i=1}^c w_i a_i \\ u_p(k) = k_p e(k) + k_D \frac{e(k) - e(k-1)}{t_s} \end{cases} \quad (27)$$

where w_i are weights of the CMAC, a_i are binary vectors, $u_n(k)$ is output of the CMAC, $u_p(k)$ is output of the PD controller, k_p and k_D are gains, t_s is the sampling time and e is the error between y and r .

And the learning function of the CMAC is given as

$$\begin{cases} E(k) = \frac{1}{2} (u(k) - u_n(k))^2 \frac{a_i}{c} \\ \Delta w(k) = \eta \frac{u(k) - u_n(k)}{c} a_i = \eta \frac{u_p(k)}{c} a_i \\ w(k) = w(k-1) + \Delta w(k) + \alpha (w(k) - w(k-1)) \end{cases} \quad (28)$$

where E is the adjusting index, η is the learning speed and $\eta \in (0, 1)$, α is the inertial variable and $\alpha \in (0, 1)$ and c is the generalizing parameter.

Once the system starts running, the controller initializes $w = 0$, then we have $u_n = 0$ and $u = u_p$ at the beginning. At the early stage, the system is mainly controlled by the PD controller. As the CMAC is learning, error e will decrease, which makes $u_n(k)$ increase to become the main control command. Although the CMAC is trained by the output of the PD controller, the output of the CMAC is not a simple imitation of the output of the PD controller. The PD controller helps the CMAC to improve its performances, restrain disturbance and enhance stability of the closed-loop system. If the PD controller works alone, control performances are determined by the gain k_p to a great extent. If the CMAC and PD controller work together, control performances will be independent of k_p , which is flexible in a rational span.

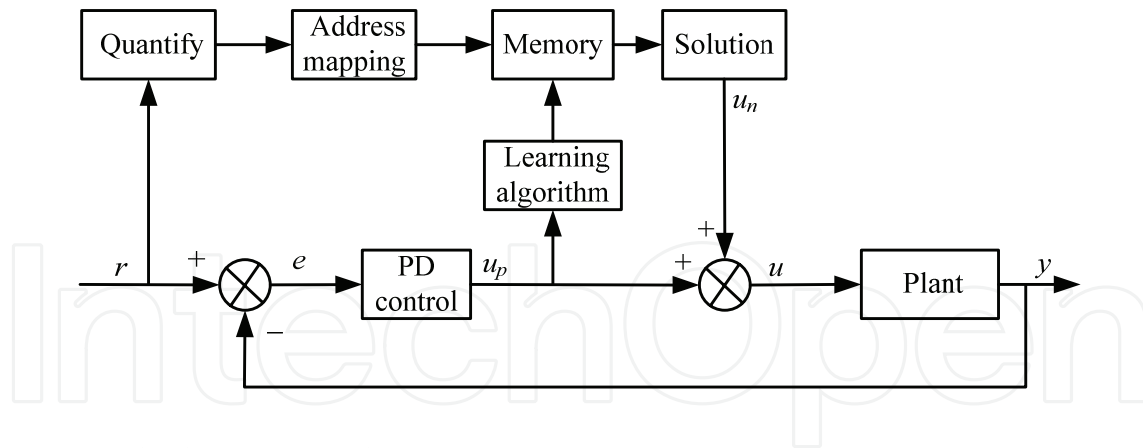


Figure 5. CMAC feedforward plus PD feedback control block diagram.

4.2. Simulations

Choose the identified model (26) as the simulation plant to tune the controller and verify its performance. The force control is aimed at tracking a ramp command, of which the expression is given as $r(t) = -10t$, where t is from 0 to 20 s.

The CMAC plus PD controller is tuned and trained by simulations. Figure 6 demonstrates that the PD controller plays a main role of control at the beginning. Then the control effect of the CMAC will gradually increase via learning from the output of the PD controller. Owing to the help of the CMAC, control performances are better than those of the only PD controller. Overshoots are reduced dramatically and control actions are speeded up while choosing the ramp signal as an input. Figure 7(a) shows an overview of control performances

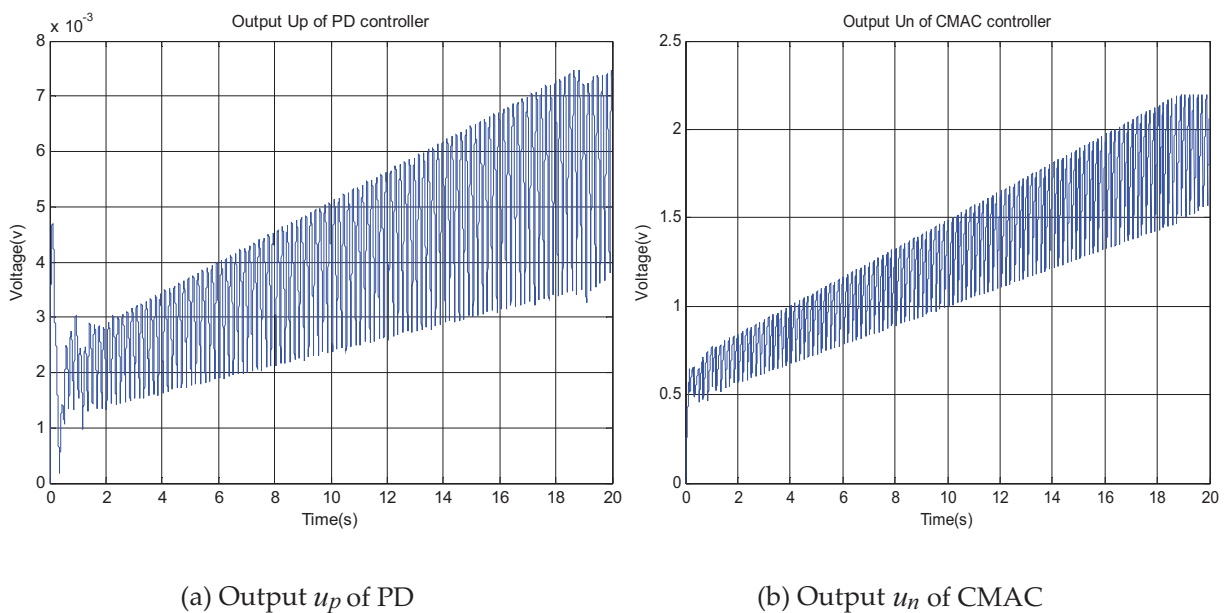


Figure 6. Output of the controller.

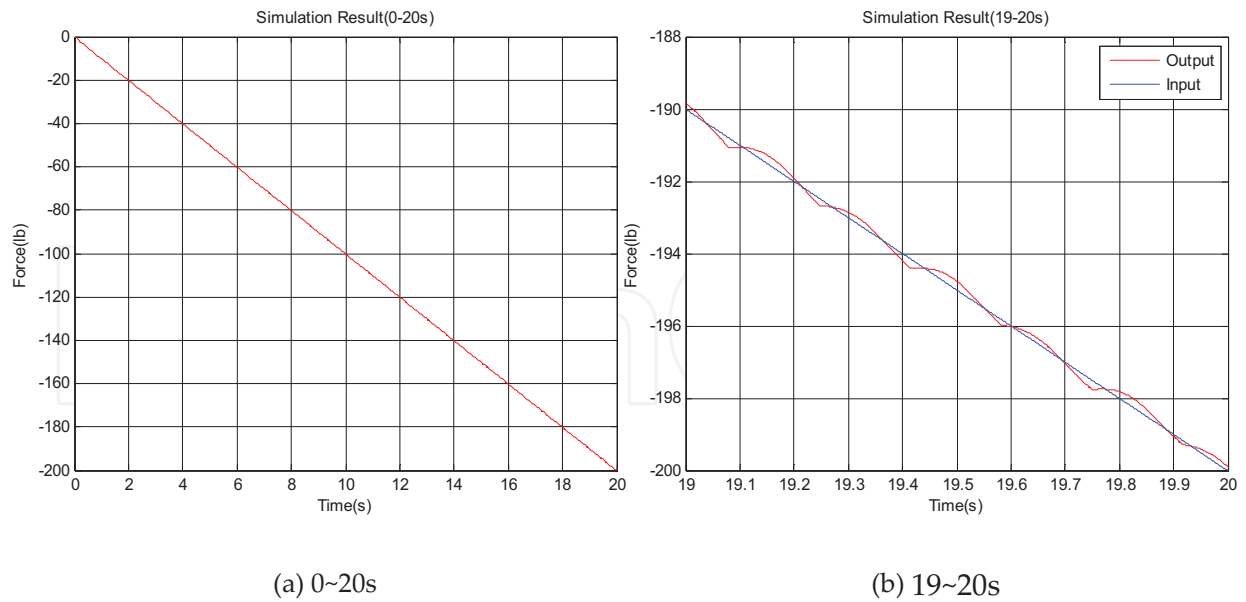


Figure 7. Simulation results.

from 0 to 20 s. **Figure 7(b)** is an enlarged view of control performances from 19 to 20s. It can be seen that the system output well tracks the input command. **Figure 8** further exhibits the superiority of the CMAC: the tracking errors are tiny, mostly between -0.2 and 0.2 lbs. Since there exists a modeling error between identified system and actual controlled plant, the controller's performances will be further validated in the real material-strength testing experiments.

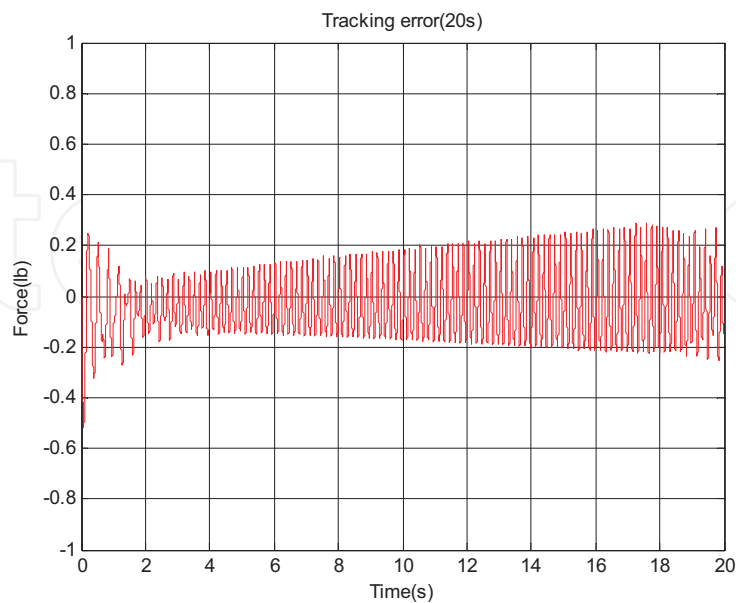


Figure 8. Tracking error.

5. Embedded system-based experiments

5.1. Experiments

For verifying the controller's adaption to modeling error in the real-time control experiments, we implement above control law in C language and download compiled files into the embedded system. The central processing unit is an ARM7-based processor LPC2294. The LPC2294 is a 32-bit reduced instruction set computer (RISC) processor with low power consumption and high performance. Although there is no Float Point Unit (FPU) in this processor, the 70 million instructions per second (MIPS) processing speed makes it ideal for the real-time control system. The control step size is set as 50 ms.

The actual force tracking control diagram is shown in **Figure 9**. The force is applied to the glass sheet specimen by a loader which is connected to a linear motor actuator through a steel bar. A three point bending test is utilized under this configuration. The load force followed a ramp function with time as the independent variable. The slope of the ramp function can be programmed through the LCD touch screen. The force range can be from 0 to 300 lbs. Displacement of the motor, which also reflected the deformation of the glass sheet specimen, is recorded during the test. The signal flow of the control system is as follows. The DAC generates the control outputs as a voltage signal, which is amplified by a power amplifier and exerted on the linear motor. The linear motor then transforms the voltage signal into rotation, and generates linear displacements. The force sensor and conditioner measure the displacements and generates charges, which are transformed back to a voltage signal and fed back through ADC to the microcontroller.

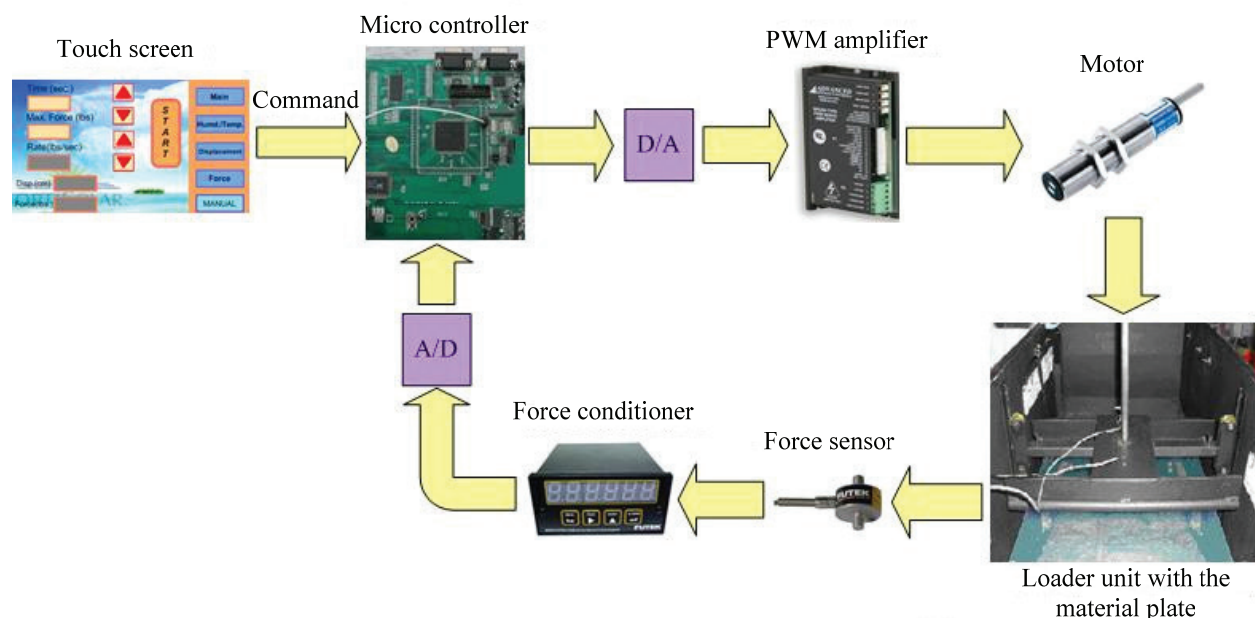


Figure 9. Actual force tracking control diagram.

The input command and control parameters are the same as those of simulations. **Figure 10(a)** shows that the output force basically tracks the input command well and the controller implemented on the embedded system can perform the control task in real time within the control step size. From **Figure 10(b)**, it can be seen that tracking errors of the experiment are mostly between -2.5 and 2.5 lbs, which are larger than those of the simulation. The reason is that uncertainties and disturbances always exist in the real world, which mainly reflects in modeling error. However, the maximum absolute tracking error is 3.79 , and the variance of tracking errors is 1.69 , which are still acceptable in the actual real-time control environment. Moreover, it verifies that the controller tuned by simulations is also available for actual experiments, and indirectly proves the effectiveness of identified system model.

5.2. Comparisons

As shown in **Figure 11**, an inverse model feedforward control is compared with the CMAC feedforward control by embedded-system-based experiments. Its basic idea is to directly employ the inverse model $1/G(s)$ of the plant's identified transfer function $G(s)$ in Eq. (26) to be the feedforward part. Obviously, this control scheme is also suitable for real-time implementations; even its computation cost is less than that of the CMAC scheme.

Figure 12(a) shows overall performances of the tracking control with the inverse model scheme. It can be seen that the output force basically tracks the input command and the controller implemented on the embedded system can perform the control task in real time within the control step size; however, the tracking control is not performed very well. As shown in **Figure 12(b)**, tracking errors are mostly between -5 and -2 lbs, which are larger than those of the CMAC scheme. **Table 2** also shows that the maximum absolute tracking error and variance of tracking errors of the inverse model scheme are both larger than those of the CMAC scheme, so that performances of the CMAC scheme are superior to those of the inverse model

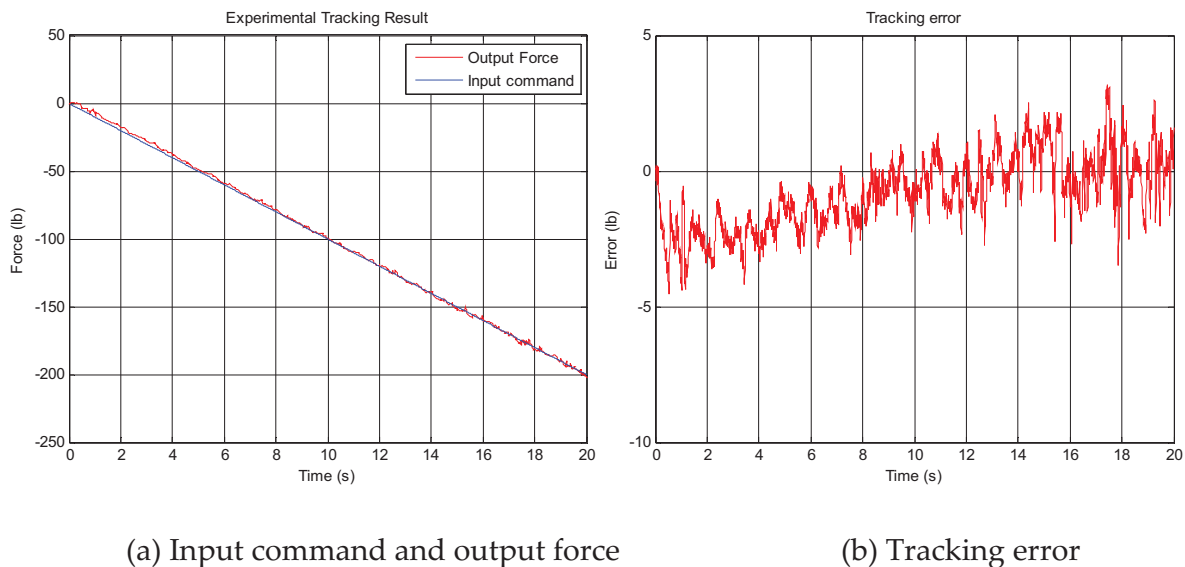


Figure 10. Experimental results.

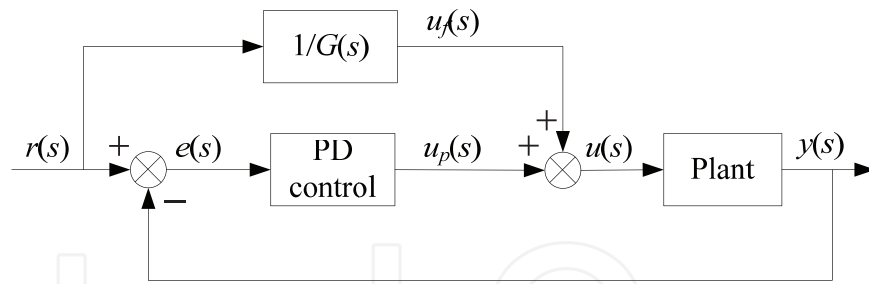


Figure 11. Block diagram of the inverse model feedforward control.

scheme. Although the CMAC scheme and inverse model scheme both utilize inverse model idea, the difference is that the latter directly employs the unchanged inverse model of the identified transfer function and the modeling error always exists, but the inverse model approximated by the CMAC is dynamical and adaptive.

5.3. Results analysis

Compared with the traditional PID control method, the learning behavior and adaptive feature of the proposed CMAC control algorithm are embodied in the freedom of tuning control parameters and the robustness to the disturbances in the real world.

The PID control method has been widely used because of its simpleness, but the tuning problem of PID parameters (proportional, integral and differential) is difficult. At present, PID parameter-tuning optimization depends on the experiences of technical staffs and needs a

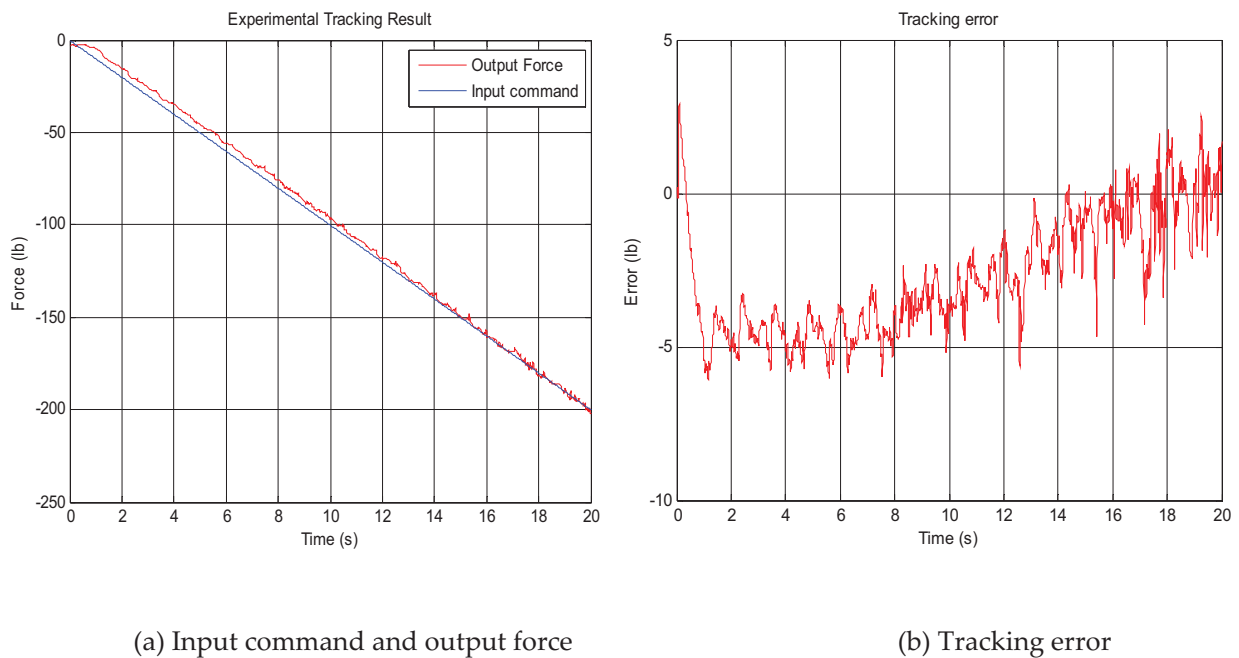


Figure 12. Experimental results (inverse model feedforward control).

Method	Maximum absolute error	Variance
CMAC	3.79	1.69
Inverse model	6.13	3.85

Table 2. Comparisons between the CMAC and the inverse model.

lot of manpower and time, which means that the optimal PID parameters are difficult to be obtained by people's tuning, and the inappropriate parameters cannot guarantee the control performances to meet the control requirements. In addition, the PID control law is a kind of the linear control law which owns few robustness to the disturbances. It means that even though the PID parameters are tuned optimally and perfectly in the simulations, the tuned parameters may perform poor in the real world due to disturbances and uncertainties.

Thus, combining the adaptive CMAC and the traditional PID to construct an intelligent neural network PID controller, can automatically identify the controlled plant and adaptively adapt the control parameters of the CMAC, which can solve the difficult problem of tuning parameters of the traditional PID controller. As shown in Sections 4 and 5, the PID parameters of the proposed CMAC algorithm in the experiments are same as those in the simulations, which verifies control performances of the proposed CMAC algorithm are independent of tuning PID parameters. Experimental results in the real world in Section 5 also demonstrate the robustness of the proposed algorithm owing to the CMAC neural network, while the traditional PID controller does not have this capacity.

6. Conclusions

In this chapter, a three-step systematic design approach is proposed to design an adaptive control system for practical use. We firstly study the system model identification problem of the embedded control material-strength testing system, including mathematical modeling of all the open-loop physical components and parameters identification of the mathematical model. Both theoretical analyses and experimental comparisons validate the identified transfer function of the system model is applicable for controller design and simulation. Next, benefited from limited computation cost and compensation ability to the modeling error, a simple and effective CMAC plus PD controller is simulated based on the identified system model, and then applied to the embedded control system for real-time force tracking. Both numerical simulations and actual experiments illustrate the proposed algorithm satisfactorily performs the tracking control task under real-time constraints of the embedded system.

On the other hand, different strength features of different types of the material plates will affect the control performances. Since the yield strength generated by the solar panel glass has an almost linear relationship with the deformation of the glass sheet, K_f in Eq. (7) is a constant. If the tested material plate is not a solar panel glass and the yield strength generated does not have a linear relationship with the deformation of the material plate, K_f in Eq. (7) will not be a

constant. In this case, the problems how to identify the parameters of the system model accurately for simulation-based controller design and how to adjust the CMAC plus PD controller to compensate the varying K_p will be under consideration in our future work.

Acknowledgements

This work is supported by National Key R&D Program of China under Grant Nos. 2016YFD0200700 and 2017YFD0701000, and Chinese Universities Scientific Fund under Grant Nos. 2017QC139 and 2017GX001.

Author details

Jian Chen^{1,2*}, Peng Li³, Gangbing Song³, Shubo Wang^{1,2}, Zichao Zhang^{1,2}, Guangqi Wang^{1,2}, Yu Tan^{1,2} and Yongjun Zheng^{1,2}

*Address all correspondence to: jchen@cau.edu.cn

1 College of Engineering, China Agricultural University, Beijing, China

2 Key Laboratory of Soil-Machine-Plant System Technology, Ministry of Agriculture, Beijing, China

3 Department of Mechanical Engineering, University of Houston, TX, USA

References

- [1] Prasad D, Snow M. Designing with Solar Power: A Source Book for Building Integrated Photovoltaics (BiPV). New York: Routledge; 2014
- [2] Dhere NG, Raravikar NR. Adhesion shear strength and surface analysis of a PV module deployed in harsh coastal climate. *Solar Energy Materials & Solar Cells*. 2001;**67**(1):363-367
- [3] Burrows K, Fthenakis V. Glass needs for a growing photovoltaics industry. *Solar Energy Materials & Solar Cells*. 2015;**132**:455-459
- [4] Hughes ZM, Pont MJ. Reducing the impact of task overruns in resource-constrained embedded systems in which a time-triggered software architecture is employed. *Transactions of the Institute of Measurement and Control*. 2008;**30**(5):427-450
- [5] Huang SJ, Yu CK, Lin JY. Intelligent robotic impedance control using embedded system structure. *Transactions of the Institute of Measurement and Control*. 2012;**35**(5):561-573
- [6] Moallem P, Zargari A, Kiyoumars A. Improving IEC flickermeter for implementation by an ARM microcontroller-based digital system. *Transactions of the Institute of Measurement and Control*. 2013;**35**(3):342-352

- [7] Albus JS. A new approach to manipulator control: The cerebellar model articulation controller (CMAC). *Journal of Dynamic Systems, Measurement, and Control*. 1975;**97**(3):220-227
- [8] Chiang CT, Lin CS. CMAC with general basis functions. *Neural Networks*. 1996;**9**(7):1199-1211
- [9] Lewis FL, Jagannathan S, Yesildirak A. *Neural Network Control of Robot Manipulators and Non-linear Systems*. Boca Raton: CRC Press; 1998
- [10] Yang B, Bao R, Han H. Robust hybrid control based on PD and novel CMAC with improved architecture and learning scheme for electric load simulator. *IEEE Transactions on Industrial Electronics*. 2014;**61**(10):5271-5279
- [11] Yang B, Han H. A CMAC-PD compound torque controller with fast learning capacity and improved output smoothness for electric load simulator. *International Journal of Control, Automation and Systems*. 2014;**12**(4):805-812
- [12] Commuri S, Lewis FL. CMAC neural networks for control of nonlinear dynamical systems: Structure, stability and passivity. *Automatica*. 1997;**33**(4):635-641
- [13] Jagannathan S, Commuri S, Lewis FL. Feedback linearization using CMAC neural networks. *Automatica*. 1998;**34**(5):547-557
- [14] Jagannathan S, Lewis FL. Robust backstepping control of robotic systems using neural networks. *Journal of Intelligent and Robotic Systems*. 1998;**23**(2-4):105-128
- [15] Kim YH, Lewis FL. Intelligent optimal design of CMAC neural network for robot manipulators. In: *Soft Computing for Intelligent Robotic Systems*. New York: Physica-Verlag HD; 1998
- [16] Kim YH, Lewis FL. Optimal design of CMAC neural-network controller for robot manipulators. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*. 2000;**30**(1):22-31
- [17] Nie J, Linkens DA. FCMAC: A fuzzified cerebellar model articulation controller with self-organizing capacity. *Automatica*. 1994;**30**(4):655-664
- [18] Wen CM, Cheng MY. Development of a recurrent fuzzy CMAC with adjustable input space quantization and self-tuning learning rate for control of a dual-axis piezoelectric actuated micromotion stage. *IEEE Transactions on Industrial Electronics*. 2013;**60**(11):5105-5115
- [19] Lee CH, Chang FY, Lin CM. An efficient interval type-2 fuzzy CMAC for chaos time-series prediction and synchronization. *IEEE Transactions on Cybernetics*. 2014;**44**(3):329-341
- [20] Lin CM, Li HY. Intelligent control using the wavelet fuzzy CMAC backstepping control system for two-axis linear piezoelectric ceramic motor drive systems. *IEEE Transactions on Fuzzy Systems*. 2014;**22**(4):791-802
- [21] Lin CM, Li HY. Adaptive dynamic sliding-mode fuzzy CMAC for voice coil motor using asymmetric gaussian membership function. *IEEE Transactions on Industrial Electronics*. 2014;**61**(10):5662-5671

- [22] Lin FJ, Yang KJ, Sun IF, Chang JK. Intelligent position control of permanent magnet synchronous motor using recurrent fuzzy neural cerebellar model articulation network. *IET Electric Power Applications*. 2015;**9**(3):248-264
- [23] Lin CM, Peng YF, Hsu CF. Robust cerebellar model articulation controller design for unknown nonlinear systems. *IEEE Transactions on Circuits and Systems II: Express Briefs*. 2004;**51**(7):354-358
- [24] Lin CM, Peng YF. Adaptive CMAC-based supervisory control for uncertain nonlinear systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*. 2004;**34**(2):1248-1260
- [25] Lin CM, Peng YF. Missile guidance law design using adaptive cerebellar model articulation controller. *IEEE Transactions on Neural Networks*. 2005;**16**(3):636-644
- [26] Lin CM, Chen TY. Self-organizing CMAC control for a class of MIMO uncertain nonlinear systems. *IEEE Transactions on Neural Networks*. 2009;**20**(9):1377-1384
- [27] Hu Y, Zeng L, Ma D. *Theory and Design of Servo Systems*. Beijing: Beijing Institute of Technology Press; 1993
- [28] Robet P, Gautier M, Bergmann C. A frequency approach for current loop modeling with a PWM converter. *IEEE Transactions on Industry Applications*. 1998;**34**(5):1003-1014
- [29] Levy EC. Complex-curve fitting. *IEEE Transactions on Automatic Control*. 1959;**4**(1):37-43

