Supervisory intelligent control system design for forward DC–DC converters

C.-F. Hsu, C.-M. Lin and K.-H. Cheng

Abstract: A supervisory intelligent control system is developed. The supervisory intelligent control system is comprised of a neural controller and a supervisory controller. The neural controller is investigated to mimic an ideal controller and the supervisory controller is designed to compensate for the approximation error between the neural controller and the ideal controller. In the proposed control scheme, an online parameter training methodology is developed based on the gradient descent method and the Lyapunov stability theorem, so that the control system can guarantee system stability. Finally, to investigate the effectiveness of the proposed control scheme, it is applied to control a forward DC–DC converter. A comparison between a PI controller, a fuzzy controller, a fuzzy neural network controller and the supervisory intelligent controller is made. Experimental results show that the proposed control system can achieve favourable regulation performances even for different input voltages and under load resistance variations.

1 Introduction

DC–DC converters are power electronic systems that convert one level of electrical voltage into another level by switching action [1, 2]. They can be used extensively in personal computers, computer peripherals and adapters of consumer electronic devices to provide DC voltages. From the control viewpoint, the controller design of the DC–DC converter is an intriguing issue owing to its intrinsic nonlinearity, which must cope with a wide input voltage and load resistance variations to ensure stability in any operating condition while providing fast transient response. For many years, the controller design was limited to PI control [3–5]. The selection of the controller parameters is a tradeoff between robustness and fast transient response. In general, it induces an overshoot in output voltage as the rise time of response is reduced. Recently, several approaches have been addressed by using sliding-mode control techniques [6–8] and fuzzy control techniques [7, 9, 10] for DC–DC converters. However, most of these approaches require time-consuming trial-and-error tuning procedures to achieve satisfactory performance; some of them cannot achieve satisfactory performance under change of operating point; and some of them do not give the stability analysis.

The neural-network-based control technique has represented an alternative design method for identification and control of these systems [11–15]. The successful key element is the approximation ability, where the parameterised neural network can approximate the unknown system dynamics of the ideal controller after learning. Recently, the concept of incorporating fuzzy logic into a neural network has grown into a popular research topic [11]. The fuzzy neural network possesses the advantages of both fuzzy systems and neural networks since it combines fuzzy reasoning capability and neural network online learning capability. The fuzzy neural network has been widely adopted for control of complex dynamical systems owing to its fast learning property and good generalisation capability compared with the neural network [16–20]. These online learning algorithms are based on the gradient descent method [16, 18], the Lyapunov stability theorem [17, 19], and the genetic algorithm [20]. So the stability, convergence and robustness of the neural-network-based control system can be improved. For real-time applications, the basic issue of neural-network-based control techniques is to provide an online learning algorithm that does not require preliminary offline tuning.

The motivation of this paper is to design a supervisory intelligent control system using the fuzzy neural network approach and the Lyapunov stability technique for the DC–DC converter. In the supervisory intelligent control system, a neural controller is utilised as a main controller, in which the interconnection weights of the fuzzy neural network are tuned online in the sense of the gradient descent method; and a supervisory controller is designed to guarantee the system stability in the sense of the Lyapunov stability theorem. Finally, to investigate the effectiveness of the proposed supervisory intelligent control scheme, it is applied to control a forward DC–DC converter. A comparison between a PI controller, a fuzzy controller, a fuzzy neural network controller and the proposed supervisory intelligent controller is made. Experimental results show that the proposed control algorithm can achieve favourable responses, including fast rise time and settling time, and small overshoot even for different input voltages and under load resistance variations. Thus, the supervisory intelligent control is more suitable to control forward DC–DC converters since a self-learning scheme is applied.

2 PWM DC–DC converter

The switch-mode DC–DC converter can convert one level of electrical voltage into another level by switching action. Nowadays, it is very popular because of its high efficiency and small size. In switch-mode DC–DC converters, power

IEE Proceedings online no. 20050376

doi:10.1049/ip-epa:20050376

Paper first received 11th April 2005 and in final revised form 8th March 2006

C.-F. Hsu is with the Department of Electrical and Control Engineering, National Chiao-Tung University, Hsinchu, 300, Taiwan, Republic of China

C.-M. Lin is with the Department of Electrical Engineering, Yuan-Ze University, Chung-Li, Tao-Yuan, 320, Taiwan, Republic of China

K.-H. Cheng is with the Department of Electrical Engineering, Chang Gung University, Tao-Yuan, 333, Taiwan, Republic of China

E-mail: fei@cn.nctu.edu.tw

switches cut off the load current within the turn-on and turn-off times under switching conditions. The output voltage is controlled by adjusting the on time of the power switch, which in turn adjusts the width of a voltage pulse at the output. This is known as pulse-width modulator (PWM) control, where the switch frequency is constant and the duty cycle, $d(N)$, varies with load resistance variations at the Nth sampling time. The output of the designed controller, $\delta d(N)$, is the change of the duty cycle. Then, the duty cycle is determined by adding the change of duty cycle $\delta d(N)$ to the previous duty cycle $d(N-1)$, i.e.

$$d(N) = d(N-1) + \delta d(N) \tag{1}$$

This duty cycle signal is then sent to a PWM output stage that generates the appropriate switching pattern for the switch in the DC–DC converter. In this paper, a widely used forward DC–DC converter is discussed and is shown in Fig. 1, where V_i and V_o are the input and output voltages of the converter, respectively, and assume V_i is assumed to be a constant voltage; D_1 and D_2 are the diodes; L is the inductor, C is the output capacitor; and Q is the transistor which controls the converter circuit operating in different modes. When the transistor is on, V_i appears across the primary and then generates

$$V_x = \frac{N_m}{N_P}(V_i - V_{lost}) \tag{2}$$

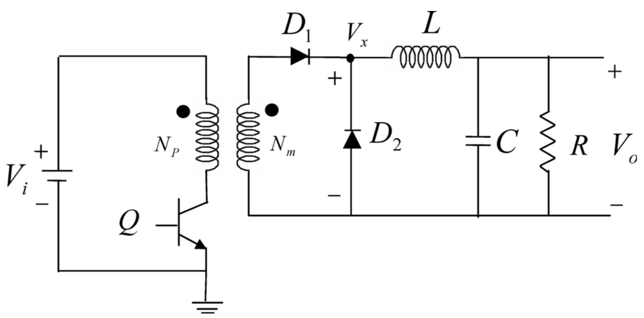


Fig. 1 *Forward DC–DC converter*

where V_{lost} is the voltage drop occurring by transistor and diodes, and represents the unmodelled dynamics in practical applications. The diode D_1 on the secondary ensures that only positive voltages are applied to the output circuit while diode D_2 provides a circulating path for inductor current if the transformer voltage is zero or negative. By the averaging method, the output voltage can be expressed as [2]

$$V_o(N) = \frac{N_m}{N_P}(V_i - V_{lost})d(N) \tag{3}$$

where N_P is the turns of the primary power winding and N_m is the turns of the slave power winding. The control problem of the forward DC–DC converter is to control the duty cycle so that the output voltage $V_o(N)$ can provide a fixed voltage under the occurrence of the uncertainties such as different input voltages and load resistance variations. The output error voltage is defined as

$$e(N) = V_o(N) - V_{ref} \tag{4}$$

where V_{ref} is the reference output voltage. The control law of the duty cycle is determined by the error voltage signal to provide fast transient response and small overshoot in the output voltage.

3 Supervisory intelligent controller design

The block diagram of the supervisory intelligent control for the power electronic system is shown in Fig. 2, in which the control law is taken as

$$\delta d_{si} = \delta d_{nc} + \delta d_{sc} \tag{5}$$

where the neural controller δd_{nc} is investigated to mimic an ideal controller and the supervisory controller δd_{sc} is designed to compensate for the approximation error. The inputs of the neural controller are the output error voltage e and its derivative, and the input of the supervisory controller is the tracking index, which is defined as

$$s = e + \lambda \int_0^t e d\tau \tag{6}$$

where λ is a positive constant.

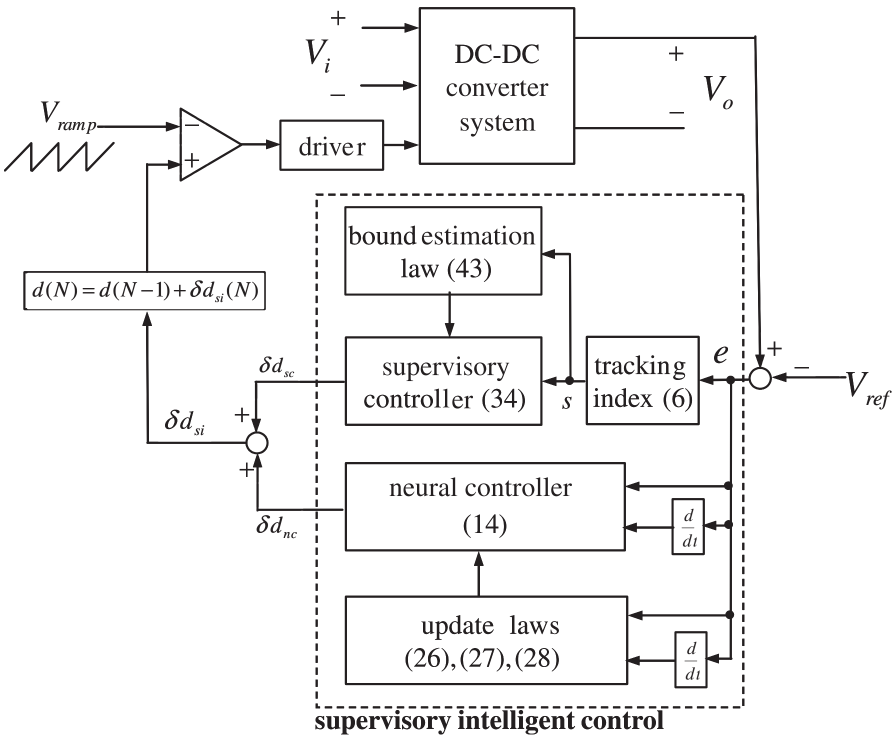


Fig. 2 *Block diagram of supervisory intelligent control for DC–DC converters*

3.1 Description of neural controller

A four-layer fuzzy neural network is shown in Fig. 3, which comprises the input (the i layer), membership (the j layer), rule (the k layer), and output (the o layer) layers. Layer 1 accepts the input variables. Layer 2 is used to calculate the Gaussian membership values. The nodes of layer 3 represent the fuzzy rules. The links before layer 3 represent the preconditions of the rules, and the links after layer 3 represent the consequences of the rule nodes. Layer 4 is the output layer. The node in this layer is the output of the fuzzy neural network. The interactions for the layers are given as follows [11, 16]:

Layer 1, Input layer: For every node i in this layer, the net input and the net output are represented as

$$net_i^1 = x_i^1 \tag{7}$$

$$y_i^1 = f_i^1(net_i^1) = net_i^1, \quad i = 1, 2 \tag{8}$$

where x_i^1 represents the i-th input to the node of layer 1.

Layer 2, Membership layer: In this layer, each node performs a membership function and acts as an element for membership degree calculation, where the Gaussian function is adopted as the membership function. For the jth node

$$net_j^2 = -\frac{(x_i^2 - m_{ij}^2)^2}{(\sigma_{ij}^2)^2} \tag{9}$$

$$y_j^2 = f_j^2(net_j^2) = \exp(net_j^2), \quad j = 1, 2, \ldots, l \tag{10}$$

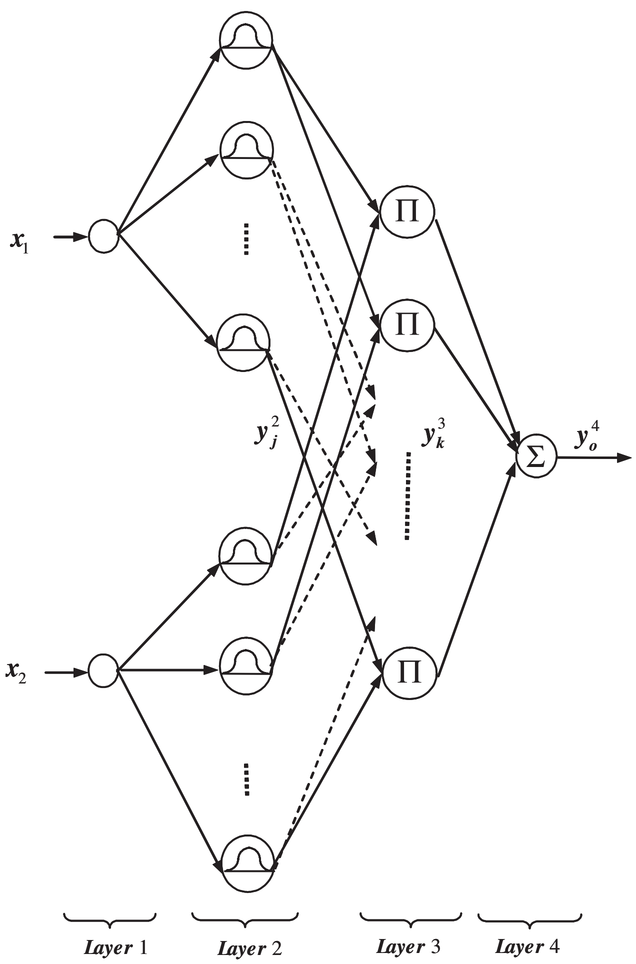


Fig. 3 *Structure of fuzzy neural network*

where m_{ij}^2 and σ_{ij}^2 are the mean and standard deviation of the Gaussian function in the jth term of the ith input linguistic variable x_i^2 to the node of layer 2, respectively.

Layer 3, Rule layer: Each node k in this layer is denoted by Π, which multiplies the incoming signals and outputs the result of the product. For the kth rule node

$$net_k^3 = \prod_j w_{jk}^3 x_j^3 \tag{11}$$

$$y_k^3 = f_k^3(net_k^3) = net_k^3, \quad k = 1, 2, \ldots, n \tag{12}$$

where x_j^3 represents the jth input to the node of layer 3, and w_{jk}^3 are the weights between the membership layer and the rule layer, which are assumed to be unity.

Layer 4, Output layer: The single node o in this layer is labelled as Σ, which computes the overall output as the summation of all incoming signals:

$$net_o^4 = \sum_k w_k^4 x_k^4 \tag{13}$$

$$y_o^4 = f_o^4(net_o^4) = net_o^4 \tag{14}$$

where the link weight w_k^4 is the output action strength associated with the kth rule, x_k^4 represents the kth input to the node of layer 4, and y_o^4 is the output of the fuzzy neural network.

3.2 Online learning algorithm

The central part of the learning algorithm for the neural controller concerns how to recursively obtain a gradient vector, which is defined as the derivative of an energy function with respect to a parameter of the neural network using the chain rule [10]. To describe the online learning algorithm of the fuzzy neural network, first the energy function $_E$ is defined as:

$$E = \frac{1}{2} e(N)^2 \tag{15}$$

Then, the learning algorithm based on the gradient descent method is described below [11, 16].

Layer 4:

$$\delta_o^4 = -\frac{\partial E}{\partial net_o^4} = -\frac{\partial E}{\partial y_o^4} \frac{\partial y_o^4}{\partial net_o^4} \tag{16}$$

The Jacobian term of the plant, $\partial E / \partial y_o^4$, can be expressed as

$$\begin{aligned}
\frac{\partial E}{\partial y_o^4} &= \frac{\partial \frac{1}{2}(V_0(N) - V_{ref})^2}{\partial \delta d_{nc}(N)} \\
&= e(N) \frac{\partial V_o(N)}{\partial \delta d_{nc}(N)} \\
&= \frac{N_m}{N_P}(V_i - V_{lost}) e(N) \frac{\partial d(N)}{\partial \delta d_{nc}(N)} \\
&= \frac{N_m}{N_P}(V_i - V_{lost}) e(N) \frac{\partial(d(N-1) + \partial d_{si}(N))}{\partial \delta d_{nc}(N)} \\
&= \frac{N_m}{N_P}(V_i - V_{lost}) e(N)
\end{aligned} \tag{17}$$

and the weight is updated by an amount

$$\Delta w_k^4 = -\eta_w \frac{\partial E}{\partial w_k^4} = \left[-\eta_w \frac{\partial E}{\partial y_o^4}\frac{\partial y_o^4}{\partial net_o^4}\right]\left[\frac{\partial net_o^4}{\partial w_k^4}\right]$$
$$= -\eta_w \delta_o^4 x_k^4 \tag{18}$$

where η_w is the learning rate of the connecting weights of the fuzzy neural network. The weights of the output layer are updated according to the following:

$$w_k^4(N+1) = w_k^4(N) + \Delta w_k^4 \tag{19}$$

Layer 3: Since the weights in this layer are unities, only the error term needs to be calculated and propagated:

$$\delta_k^3 = -\frac{\partial E}{\partial net_k^3} = \left[-\frac{\partial E}{\partial y_o^4}\frac{\partial y_o^4}{\partial net_o^4}\right]\left[-\frac{\partial net_o^4}{\partial y_k^3}\frac{\partial y_k^3}{\partial net_k^3}\right]$$
$$= \delta_o^4 w_k^4 \tag{20}$$

Layer 2: The multiplication operation is done in this layer. The error term is computed as follows:

$$\delta_j^2 = -\frac{\partial E}{\partial net_j^2} = \left[-\frac{\partial E}{\partial y_o^4}\frac{\partial y_o^4}{\partial net_o^4}\frac{\partial net_o^4}{\partial y_k^3}\frac{\partial y_k^3}{\partial net_k^3}\right]$$
$$\times \left[\frac{\partial net_k^3}{\partial y_i^2}\frac{\partial y_j^2}{\partial net_j^2}\right] = \delta_o^4 \sum_k w_k^4 y_k^3 \tag{21}$$

and the update law of m_{ij}^2 is

$$\Delta m_{ij}^2 = -\eta_m \frac{\partial E}{\partial m_{ij}^2}$$
$$= \left[-\eta_m \frac{\partial E}{\partial y_o^4}\frac{\partial y_o^4}{\partial net_o^4}\frac{\partial net_o^4}{\partial y_k^3}\frac{\partial y_k^3}{\partial net_k^3}\frac{\partial net_k^3}{\partial y_j^2}\frac{\partial y_j^3}{\partial net_j^2}\frac{\partial net_j^2}{\partial m_{ij}^2}\right]$$
$$= -\eta_m \delta_o^4 \sum_k w_k^4 y_k^3 \frac{2(x_i^2 - m_{ij}^2)}{(\sigma_{ij}^2)^2} \tag{22}$$

where η_m is the learning rate of the mean. The update law of σ_{ij}^2 is

$$\Delta \sigma_{ij}^2 = -\eta_\sigma \frac{\partial E}{\partial \sigma_{ij}^2}$$
$$= \left[-\eta_\sigma \frac{\partial E}{\partial y_o^4}\frac{\partial y_o^4}{\partial net_o^4}\frac{\partial net_o^4}{\partial y_k^3}\frac{\partial y_k^3}{\partial net_k^3}\frac{\partial net_k^3}{\partial y_j^2}\frac{\partial y_j^2}{\partial net_j^2}\frac{\partial net_j^2}{\partial \sigma_{ij}^2}\right]$$
$$= -\eta_\sigma \delta_o^4 \sum_k w_k^4 y_k^3 \frac{2(x_i^2 - m_{ij}^2)^2}{(\sigma_{ij}^2)^3} \tag{23}$$

where η_σ is the learning rate of the standard deviation. The mean and standard deviation of the hidden layer are updated as follows:

$$m_{ij}^2(N+1) = m_{ij}^2(N) + \Delta m_{ij}^2 \tag{24}$$

$$\sigma_{ij}^2(N+1) = \sigma_{ij}^2(N) + \Delta \sigma_{ij}^2 \tag{25}$$

Since N_P, N_m, V_i and V_{lost} in (17) are unavailable, these parameters in the learning algorithms can be reorganised as positive constants in practical applications. Therefore, the update laws (18), (22), and (23) can be reconstructed as follows:

$$\Delta w_k^4 = -\eta'_w e x_k^4 \tag{26}$$

$$\Delta m_{ij}^4 = -\eta'_m e \sum_k w_k^4 y_k^3 \frac{2(x_i^2 - m_{ij}^2)}{(\sigma_{ij}^2)^2} \tag{27}$$

$$\Delta \sigma_{ij}^2 = -\eta'_\sigma e \sum_k w_k^4 y_k^3 \frac{2(x_i^2 - m_{ij}^2)^2}{(\sigma_{ij}^2)^3} \tag{28}$$

where $\eta'_w = \eta_w \frac{N_m}{N_P}(V_i - V_{lost})$, $\eta'_m = \eta_m \frac{N_m}{N_P}(V_i - V_{lost})$ and $\eta'_\sigma = \eta_\sigma \frac{N_m}{N_P}(V_i - V_{lost})$. η'_w, η'_m and η'_σ can be taken as the new learning rates.

3.3 Supervisory controller

Differentiating both sides of (3) with respect to time yields

$$\dot{V}_o = \frac{N_m}{N_P}(V_i - V_{lost})\delta d \tag{29}$$

If the parameters of the converter are well known and the external disturbance is measurable, an ideal controller can be obtained as [21]

$$\delta d^* = \frac{N_P}{N_m(V_i - V_{lost})}(\dot{V}_{ref} - \lambda e) \tag{30}$$

Substituting (30) into (29), gives

$$\dot{e} + \lambda e = 0 \tag{31}$$

Since λ is a positive constant, it implies that $\lim_{t\to\infty} e = 0$. However, since the system parameters and the voltage drop may be unknown or perturbed, the ideal controller cannot be implemented. To tackle this problem, a neural network is utilised to approximate this ideal controller. By the universal approximation theorem, there exists an optimal fuzzy neural network such that [22]

$$\delta d_{nc}(\boldsymbol{w}^*) - \delta d^* = \varepsilon \tag{32}$$

where $\boldsymbol{w}^* = [w_k^{4*} \quad m_{ij}^{2*} \quad \sigma_{ij}^{2*}]^T$ is the ideal weight vector of the neural controller, and ε denotes the approximation error and is assumed to be bounded by $0 \le |\varepsilon| \le E$, where E is a positive constant. The error bound is assumed to be a constant during the observation; however, it is difficult to measure it in practical applications. Therefore, a bound estimation is developed to observe the bound of the approximation error. Define the estimation error of the bound

$$\tilde{E} = E - \hat{E} \tag{33}$$

where $\hat{E}$ is the estimated error bound. The supervisory controller is designed to compensate for the effect of approximation error and is chosen as

$$\delta d_{sc} = -\hat{E}\,\mathrm{sgn}(s) \tag{34}$$

in which sgn(.) is a sign function. By substituting (5) into (29), it is can be shown that

$$\dot{V}_o = \frac{N_m}{N_P}(V_i - V_{lost})(\delta d_{nc} + \delta d_{sc}) \tag{35}$$

After some straightforward manipulations, the error equation governing the system can be obtained through (6), (30) and (35) as follows:

$$\dot{e} + \lambda e = \frac{N_m}{N_P}(V_i - V_{lost})(\delta d_{nc} + \delta d_{sc} - \delta d^*) = \dot{s} \tag{36}$$

Define a Lyapunov function as

$$V(s, \tilde{E}) = \frac{s^2}{2} + \frac{\tilde{E}^2}{2\eta_E} \tag{37}$$

where the positive constant η_E is a learning rate. Differentiating (37) with respect to time and using (32),

(34) and (36), we obtain

$$
\begin{aligned}
\dot{V}(s,\tilde{E}) &= s\frac{N_m}{N_P}(V_i - V_{lost})(\varepsilon + \delta d_{sc}) + \frac{\tilde{E}\dot{\tilde{E}}}{\eta_E} \\
&= \frac{N_m}{N_P}(V_i - V_{lost})(\varepsilon s + \hat{E}|s|) + \frac{\tilde{E}\dot{\tilde{E}}}{\eta_E} \qquad (38)
\end{aligned}
$$

If the bound estimation law is chosen as

$$\dot{\tilde{E}} = -\dot{\hat{E}} = -\eta_E \frac{N_m}{N_P}(V_i - V_{lost})|s| \qquad (39)$$

then (38) becomes

$$
\begin{aligned}
\dot{V}(s,\tilde{E}) &= \frac{N_m}{N_P}(V_i - V_{lost})(\varepsilon s - E|s|) \\
&\le \frac{N_m}{N_P}(V_i - V_{lost})(|\varepsilon||s| - E|s|) \\
&= -\frac{N_m}{N_P}(V_i - V_{lost})(E - |\varepsilon|)|s| \le 0 \qquad (40)
\end{aligned}
$$

Since $\dot{V}(s,\tilde{E})$ is negative semi-definite, that is $V(s(t),\tilde{E}(t)) \le V(s(0),\tilde{E}(0))$, it implies that s and $\tilde{E}$ are bounded. Let function $\Omega \equiv \frac{N_m}{N_P}(V_i - V_{lost})\ (E - |\varepsilon|)s \le \frac{N_m}{N_P}(V_i - V_{lost})\ (E - |\varepsilon|)|s| \le -\dot{V}(s,\tilde{E})$, and integrate Ω with respect to time, then we obtain

$$\int_0^t \Omega(\tau)d\tau \le V(s(0),\tilde{E}(0)) - V(s(t),\tilde{E}(t)) \qquad (41)$$

Because $V(s(0),\tilde{E}(0))$ is bounded, and $V(s(t),\tilde{E}(t))$ is nonincreasing and bounded, the following result can be obtained:

$$\lim_{t\to\infty}\int_0^t \Omega(\tau)d\tau < \infty \qquad (42)$$

Also, $\dot{\Omega}(t)$ is bounded, so by Barbalat's Lemma [22], $\lim_{t\to\infty}\Omega = 0$. That is, $s \to 0$ as $t \to \infty$. Hence, the supervisory intelligent control of power electronic systems is asymptotically stable. Similarly, since N_P, N_m, V_i and V_{lost} in (39) are unavailable in practical applications, the bound estimation law (39) can be reconstructed as follows:

$$\dot{\hat{E}} = \eta'_E|s| \qquad (43)$$

where $\eta'_E = \eta_E \frac{N_m}{N_P}(V_i - V_{lost})$. The design algorithms of the supervisory intelligent control are summarised as follows:

Step 1: The output error voltage e and the tracking index s are given in (4) and (6), respectively.

Step 2: The value of neural controller δd_{nc} is the output of the fuzzy neural network as given in (14).

Step 3: The supervisory controller δd_{sc} is given in (34) with the parameter $\hat{E}$ adapted by (43).

Step 4: The duty cycle is determined by adding the change of duty cycle to the previous duty cycle as shown in (1).

Step 5: The duty cycle signal is then sent to a PWM output stage. Then, go back to Step 1.

4 Experimental results

The computer control experimental system for the forward DC–DC converter is shown in Fig. 4. A servo control card is installed in the control computer, which includes multi-channels of D/A, A/D, PIO and encoder interface circuits. The control problem is to control the duty cycle so that the output voltage can provide a fixed voltage ($V_{ref} = 10\,V$) under the occurrence of uncertainties such as different input voltages and load resistance variations. The proposed control algorithm is realised on a Pentium processor using the 'Turbo C' language. Two experimental cases are addressed: (*a*) Case 1 (the input voltage is set as $V_i = 20\,V$); (*b*) Case 2 (the input voltage is set as $V_i = 25\,V$). In both cases, some load resistance variations with step changes are tested: (i) from 20 to 4 Ω at 300 ms, (ii) from 4 to 20 Ω at 500 ms, and (iii) from 20 to 4 Ω at 700 ms. The circuit parameter values of the forward DC–DC converter are chosen as $N_P:N_m = 4:3$, $R = 20\,\Omega$, $L = 500\,\mu H$ and $C = 2200\,\mu F$. The converter runs at a switching frequency of 20 kHz and the controller runs at a sampling frequency of 1 kHz. The duty cycle is generated by a PWM IC SG1825. The generated duty cycle is directly proportional to the analogue output of the controller. To illustrate the effectiveness of the proposed design method, a comparison between a PI controller, a fuzzy controller, a fuzzy neural network controller and the proposed supervisory intelligent controller is made.

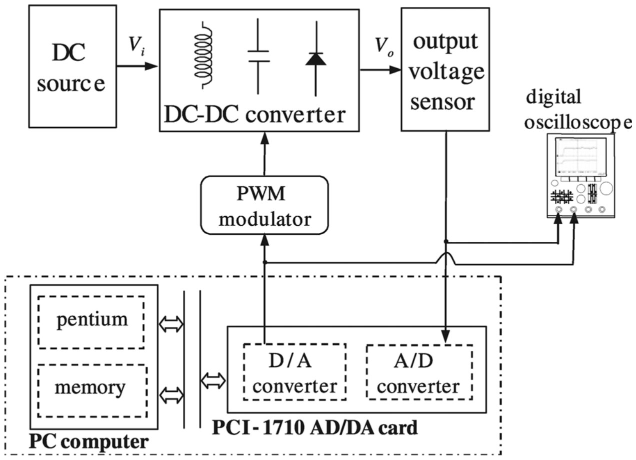


Fig. 4 *Experimental setup*

4.1 Comparison of different control method

To compare the regulation efficiency, first a PI controller proposed in [4] is applied to the forward DC–DC converter. The PI controller is given as

$$\delta d_{pi} = -0.01e - 0.2\dot{e} \qquad (44)$$

It is a PD type for the change of duty cycle δd_{pi}, therefore it is a PI type for the duty cycle d_{pi}. The experimental results for the PI controller are shown in Fig. 5. For Case 1 and Case 2, the converter responses are shown in Figs. 5*a* and *c*; and the associated control efforts are shown in Figs. 5*b* and *d*, respectively. From the experimental results, the PI controller can achieve fast tracking performances; however, there exists 5% overshoot and the PI gains are determined through a lot of trials. Next, a fuzzy controller proposed in [9] is applied to the forward DC–DC converter. The fuzzy control rules are given in the following form:

$$\text{Rule } i\text{: IF } e \text{ is } F_e^i \text{ and } \dot{e} \text{ is } F_{\dot{e}}^i\text{, THEN } \delta d_{fc} \text{ is } \rho_i \qquad (45)$$

where ρ_i, $i = 1, 2, \ldots, n$ are the singleton control actions and F_e^i and $F_{\dot{e}}^i$ are the labels of the fuzzy sets. The defuzzification of the controller output is accomplished by the method of centre-of-gravity:

$$\delta d_{fc}(N) = \frac{\sum_{i=1}^{n} v_i \times \rho_i}{\sum_{i=1}^{n} v_i} \qquad (46)$$

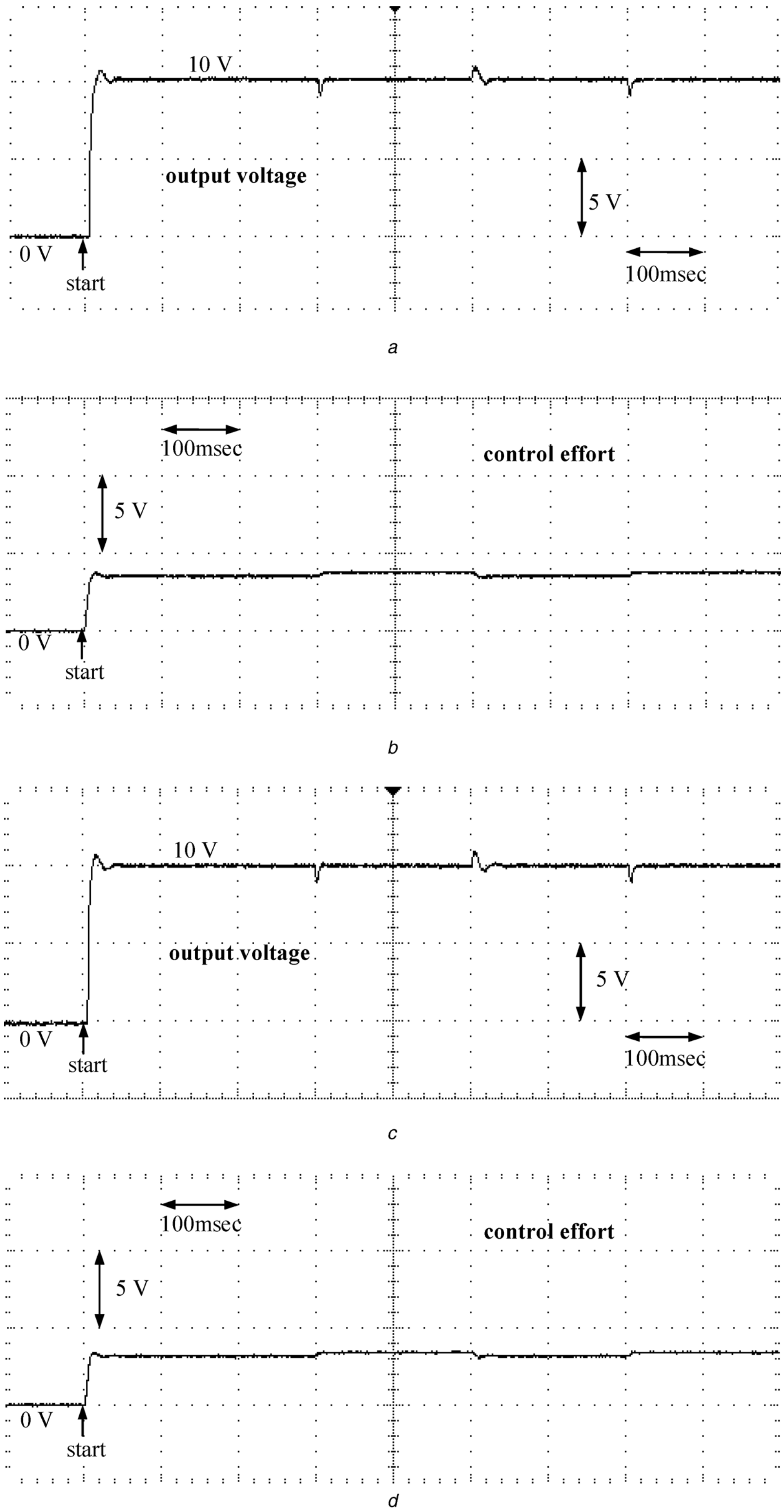
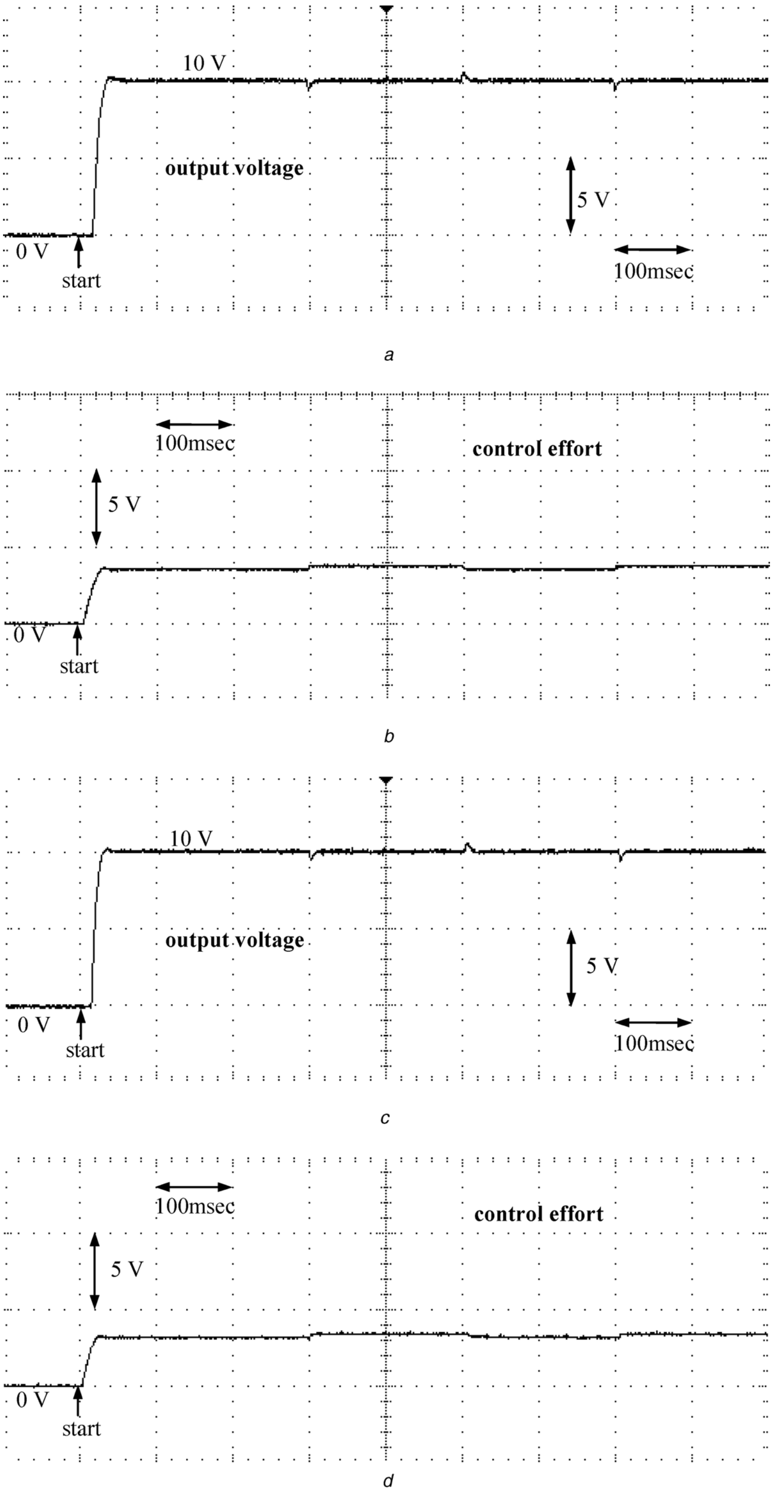


Fig. 5 *Experimental results of PI controller*
a, *b* Case 1
c, *d* Case 2

where v_i is the firing weight of the ith rule. The fuzzy rules in (45) can be constructed by the sense that e and $\dot{e}$ will approach zero with fast rise time and without large overshoot. Generally, the determination of these rules comes from human knowledge and via some trial-and-error processes. In this sense, a 25 fuzzy rule system is summarised in Table 1, where the fuzzy labels are negative big (NB), negative small (NS), zero (ZO), positive small (PS), and positive big (PB). The experimental results of the fuzzy controller for Case 1 and Case 2 are shown in Fig. 6. From the experimental results, the fuzzy controller can achieve fast tracking performance; however, the fuzzy rules base is constructed through much trial-and-error to ensure proper behaviour in the operating conditions. In the following, a fuzzy neural network controller proposed in [16] is applied to the forward DC–DC converter. The parameters of the fuzzy neural network controller are selected as $\eta'_w = \eta'_m = \eta'_\sigma = 0.001$. These parameters are

Table 1: Fuzzy rules of fuzzy controller for power electronic system

$\dot{e}$	e				
	NB	NS	ZO	PS	PB
NB	1.000	1.000	1.000	0.400	0.000
NS	1.000	1.000	0.400	0.000	– 0.400
ZO	1.000	0.400	0.000	– 0.400	– 1.000
PS	0.400	0.000	– 0.400	– 1.000	– 1.000
PB	0.000	– 0.400	– 1.000	– 1.000	– 1.000



d

Fig. 6 *Experimental results of fuzzy controller*
a, *b* Case 1
c, *d* Case 2

chosen to achieve good transient control performance considering the requirement of stability. The experimental results of the fuzzy neural network controller for Case 1 and Case 2 are shown in Fig. 7. From the experimental results, the robust tracking performance of the fuzzy neural network controller is obvious under the occurrence of the load resistance variations after training. However, the initial transient response is not good.

4.2 Supervisory intelligent control

The proposed supervisory intelligent controller is applied to the forward DC–DC converter. The parameters of the proposed controller are selected as $\lambda = 1000$, $\eta'_w = \eta'_m = \eta'_\sigma = 0.001$ and $\eta'_\varepsilon = 0.00001$; the choice of these parameters is through some trials. If the learning rates are chosen too small, then the parameter convergence of the supervisory intelligent controller will be easily achieved;

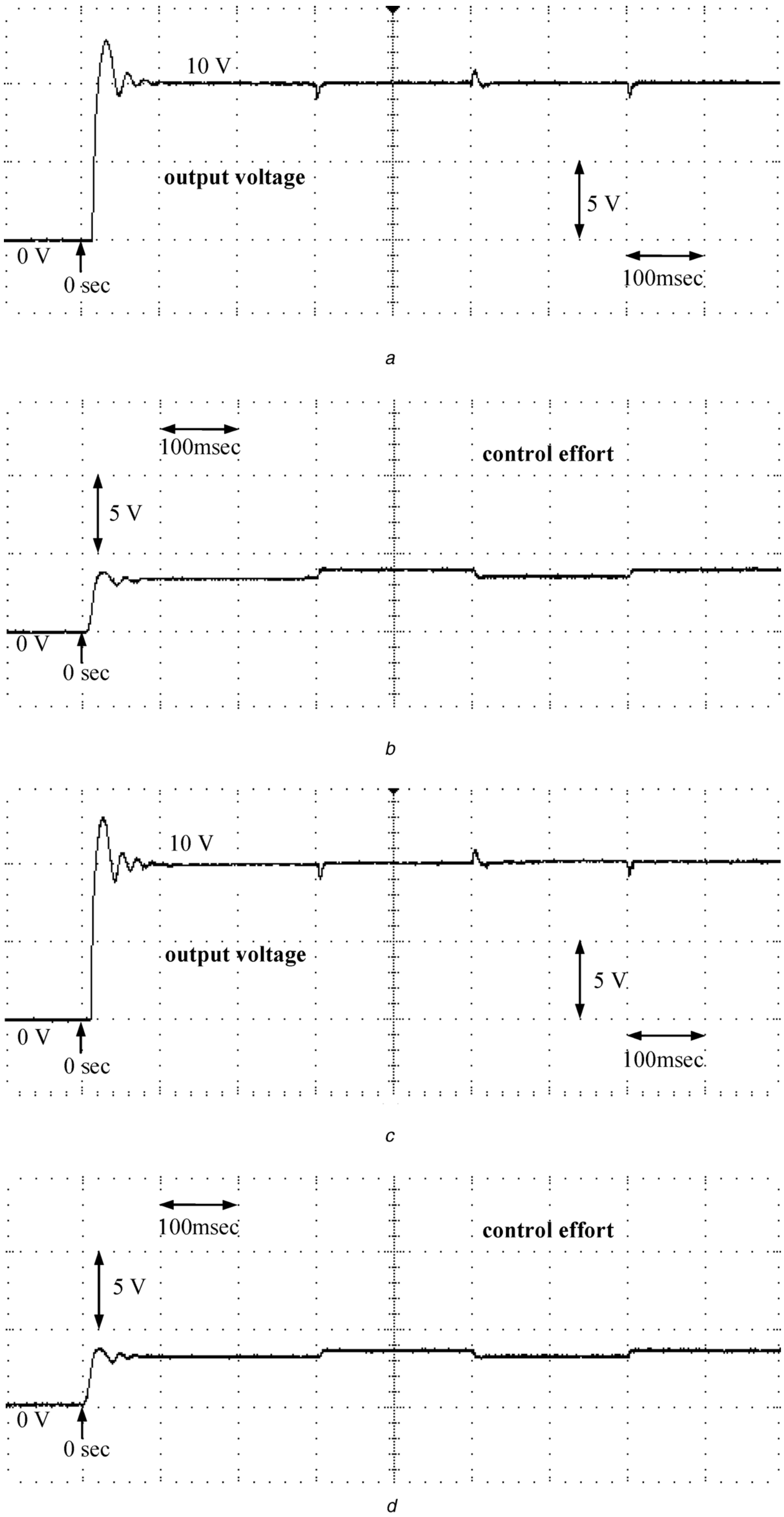


Fig. 7 *Experimental results of fuzzy neural network controller*
a, *b* Case 1
c, *d* Case 2

however, this will result in slow learning speed. On the other hand, if the learning rates are chosen too large, then the learning speed will be fast; however, the supervisory intelligent controller system may become more unstable for the parameter convergence. The experimental results of the supervisory intelligent controller for Case 1 and Case 2 are shown in Fig. 8. Since the controller parameters are initialised from zero, the supervisory intelligent controller has the drawback of large overshoot responses and control efforts at the initial learning phase. After training, the trained supervisory intelligent controller is applied to control the forward DC–DC converter system again. The experimental results of the trained supervisory intelligent controller for Case 1 and Case 2 are shown in Fig. 9. It is seen that the regulation performance of the trained supervisory intelligent controller

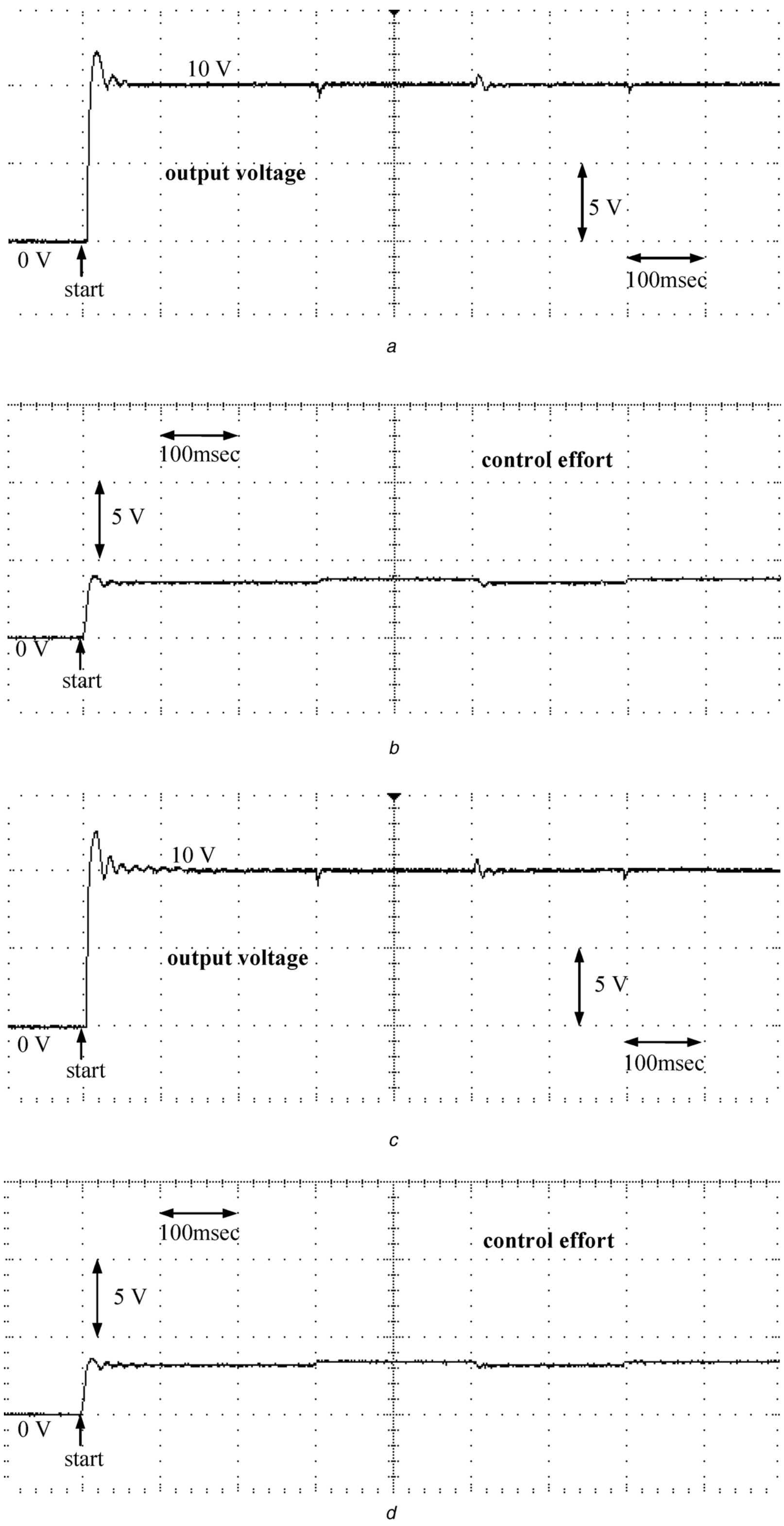


Fig. 8 *Experimental results of supervisory intelligent controller*
a, *b* Case 1
c, *d* Case 2

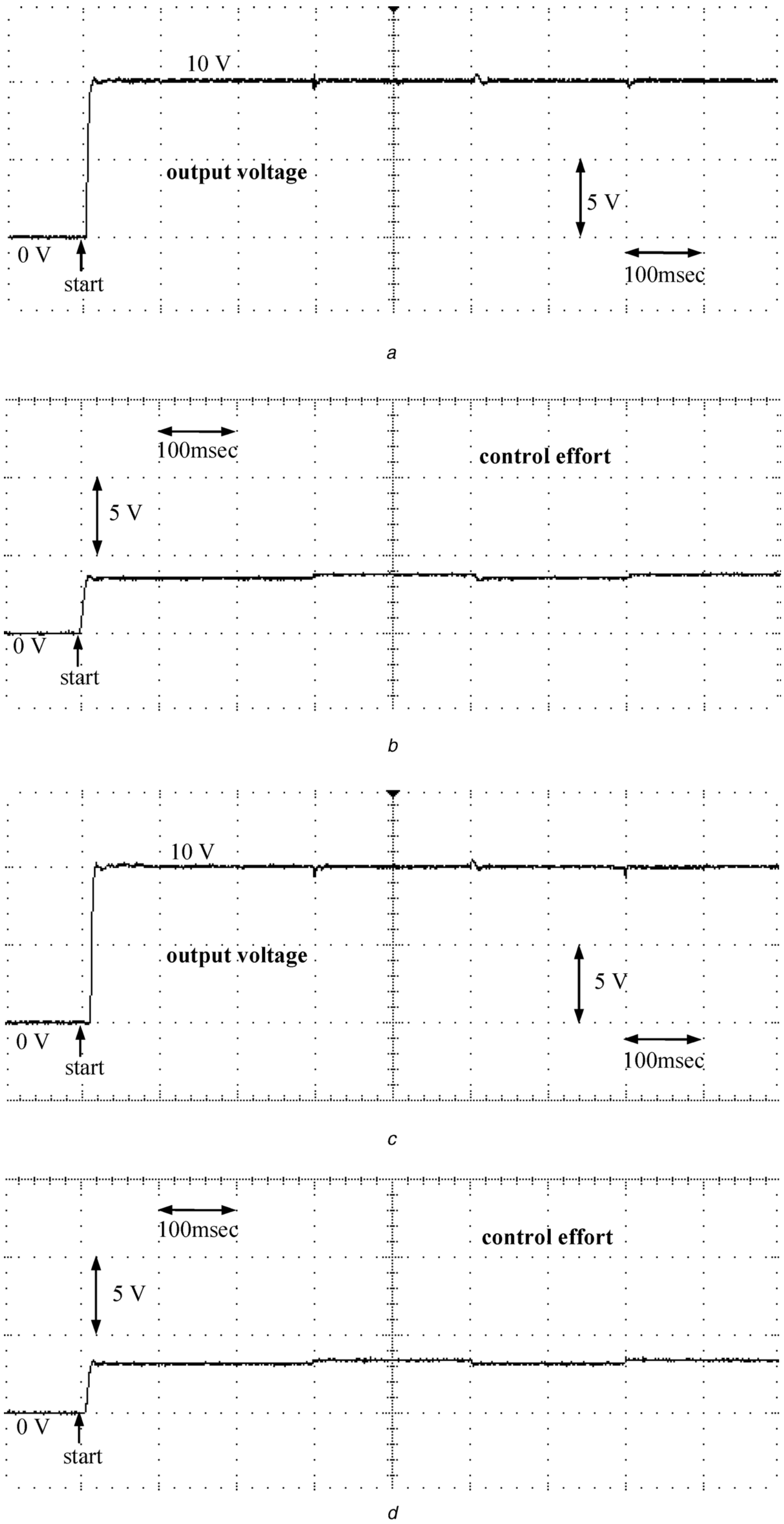


Fig. 9 *Experimental results of trained supervisory intelligent controller*
a, *b* Case 1
c, *d* Case 2

is further improved when the initial values of the controller parameters are trained. The comparisons of control performance and control characteristics for PI control, fuzzy control, fuzzy neural network control and supervisory intelligent control are summarised in Tables 2 and 3, respectively. It is seen that the supervisory intelligent control system has robust characteristics and fast transient response even for different input voltages and under load resistance variations since the online learning scheme is applied. From the view point of computation time, the supervisory intelligent control and fuzzy neural network control will pay the price of more computation time than the fuzzy control and PI control for achieving better control performance. However, the increase of computation time is acceptable for real applications.

Table 2: Performance comparison

Controller	Case 1		Case 2	
	Over-shoot (%)	Settling time (ms)	Over-shoot (%)	Settling time (ms)
PI controller	5	24	5	19
Fuzzy controller	0	38	0	34
Fuzzy neural network controller	25	54	30	56
Supervisory intelligent controller	20	48	22	44
Trained supervisory intelligent controller	0	21	0	19

Table 3: Characteristics comparison

Controller	Controller parameters	Load variation regulation ability	Stability proof	Computation time (ms)
PI controller	trial and error	middle	yes	0.180
Fuzzy controller	trial and error	good	no	0.260
Fuzzy neural network controller	online learning	excellent	no	0.295
Supervisory intelligent controller	online learning	excellent	yes	0.298
Trained supervisory intelligent controller	online learning	excellent	yes	0.298

5 Conclusions

A PI control, a fuzzy control, a fuzzy neural network control and a supervisory intelligent control have been adopted to control a forward type DC–DC converter. The proposed supervisory intelligent control system comprises a neural controller and a supervisory controller, in which the controller parameters can be tuned online based on the gradient descent method and the Lyapunov stability theorem to achieve system stability and satisfactory performance. To illustrate the effectiveness of the proposed design method, several experiments have been performed. The experimental results demonstrate the efficiency of the proposed control method. Therefore, the proposed supervisory intelligent controller is suitable for DC–DC converter control.

6 Acknowledgment

The authors appreciate the financial support in part from the National Science Council of Republic of China under grant NSC 93-2213-E-155-038. The authors would like to express their gratitude to the reviewers for their valuable comments and suggestions.

7 References

1 Middlebrook, R.D., and Cuk, S.: 'Advances in switched-mode power conversion' (Teslaco, Pasadena, CA, USA, 1981)
2 Pressman, A.I.: 'Switching power supply design' (McGraw-Hill, New York, USA, 1998)
3 Choi, B.: 'Step load response of a current-mode-controlled DC-to-DC converter', *IEEE Trans. Aerosp. Electron. Syst.*, 1997, **33**, (4), pp. 1115–1121
4 Alvarez-Ramirez, J., Cervantes, I., Espinosa-Perez, G., Maya, P., and Morales, A.: 'A stable design of PI control for DC–DC converters with an RHS zero', *IEEE Trans. Circuits Syst. I*, 2001, **48**, (1), pp. 103–106
5 Garcera, G., Figuerese, E., Pascual, M., and Benavent, J.M.: 'Robust model following control of parallel buck converters', *IEEE Trans. Aerosp. Electron. Syst.*, 2004, **40**, (3), pp. 983–997
6 Mattavelli, P., Rossetto, L., and Spiazzi, G.: 'Small-signal analysis of DC–DC converters with sliding mode control', *IEEE Trans. Power Electron.*, 1997, **12**, (1), pp. 96–102
7 Vidal-Ldiarte, E., Martine-Salamero, L., Guinjoan, F., Calvente, J., and Gomariz, S.: 'Sliding and fuzzy control of a boot converter using an 8-bit microcontroller', *IEE Proc., Electr. Power Appl.*, 2004, **151**, (1), pp. 5–11
8 Lopez, M., Vicuna, L.G., Castilla, M., Gaya, P., and Lopez, O.: 'Current distribution control design for paralleled DC/DC converters using sliding-mode control', *IEEE Trans. Ind. Electron.*, 2004, **51**, (2), pp. 419–428
9 Gupta, T., Boudreaux, R.R., Nelms, R.M., and Hung, J.Y.: 'Implementation of a fuzzy controller for dc–dc converters using an inexpensive 8-b microcontroller', *IEEE Trans. Ind. Electron.*, 1997, **44**, (5), pp. 661–669
10 Lin, P.Z., Lin, C.M., Hsu, C.F., and Lee, T.T.: 'Type-2 fuzzy controller design via sliding-mode approach for DC–DC converters', *IEE Proc., Electr. Power Appl.*, 2005, **152**, (6), pp. 1482–1488
11 Lin, C.T., and Lee, C.S.G.: 'Neural fuzzy systems: a neuro-fuzzy synergism to intelligent systems' (Prentice-Hall, Englewood Cliffs, NJ, USA, 1996)
12 Seshagiri, S., and Khalil, H.K.: 'Output feedback control of nonlinear systems using RBF neural networks', *IEEE Trans. Neural Netw.*, 2000, **11**, (1), pp. 69–79
13 Lin, C.M., and Hsu, C.F.: 'Neural-network-based adaptive control for induction servomotor drive system', *IEEE Trans. Ind. Electron.*, 2002, **49**, (1), pp. 144–156
14 Lin, C.M., and Hsu, C.F.: 'Neural-network hybrid control for antilock braking systems', *IEEE Trans. Neural Netw.*, 2003, **14**, (2), pp. 351–359
15 Hsu, C.F., Lin, C.M., and Chen, T.Y.: 'Neural-network-identification-based adaptive control of wing rock motion', *IEE Proc., Control Theory Appl.*, 2005, **155**, (1), pp. 65–71
16 Wai, R.J., and Lin, F.J.: 'Fuzzy neural network sliding-mode position controller for induction servo motor drive', *IEE Proc., Electr. Power Appl.*, 1999, **146**, (3), pp. 297–308
17 Diao, Y., and Passino, K.M.: 'Stable fault-tolerant adaptive fuzzy/neural control for a turbine engine', *IEEE Trans. Contr. Syst. Technol.*, 2001, **9**, (3), pp. 494–509
18 Chen, Y.C., and Teng, C.C.: 'A model reference control structure using a fuzzy neural network', *Fuzzy Sets Syst.*, 1995, **75**, (1), pp. 17–31
19 Lin, C.M., and Hsu, C.F.: 'Supervisory recurrent fuzzy neural network control of wing rock for slender delta wings', *IEEE Trans. Fuzzy Syst.*, 2004, **12**, (5), pp. 733–742
20 Lin, C.J.: 'A GA-based neural fuzzy system for temperature control', *Fuzzy Sets Syst.*, 2004, **143**, (2), pp. 311–333
21 Slotine, J.J.E., and Li, W.P.: 'Applied nonlinear control' (Prentice-Hall, Englewood Cliffs, NJ, USA, 1991)
22 Wang, L.X.: 'Adaptive fuzzy systems and control: design and stability analysis' (Prentice-Hall, Englewood Cliffs, NJ, USA, 1994)