

Single-neuron Quasi-PID Control Method for Switching Power Amplifiers in Protective Relay Test Devices

X.M. SUN¹, Z.F. CHENG¹, S.L. YANG², D.B. CHEN²

¹Electronics and Electrical Engineering Department, Chongqing University of Arts and Sciences, China

²Electrical Engineering, Chongqing Water Resources and Electric Engineering College, China

KEYWORD: switching power amplifiers; quasi-PID control method; single neuron; adaptive control method; protective relay test devices

ABSTRACT: Protective relay test devices (PRTDs) are indispensable tools for testing protective relays and other security and automatic equipment. To promote test qualities, it is required that modern PRTDs be able to adapt to different loads, time-variant system parameters, random disturbances and fault waveforms containing various frequency components and nonperiodic components so as to ensure the accuracies of output waveforms, and this is directly determined by the performance of power amplifiers (PAs) adopted by PRTDs. Linear PAs (LPAs) are generally the preference choices, considering the difficulties in controlling the tracking accuracies of switching PAs (SPAs). Except that, however, SPAs have many advantages over LPAs. This paper presents an effective control method, named single-neuron quasi-PID control method, to improve the tracking accuracies of SPAs to make their applications in modern PRTDs feasible. Experiments show that the mean square error of this control method is less than 0.15%, meaning that the accuracy requirement of modern PRTDs is satisfactorily fulfilled. And the computational complexity of this control method is small and so it is very suitable for realization on low-performance microprocessors.

INTRODUCTION

Protective relay test devices (PRTDs) are now indispensable in testing protective relays and other security and automatic equipment that are widely used in modern electric power systems. To meet the basic demand of testing various types of protective relays and security and automatic equipment, PRTDs are firstly required to be able to adapt to different input characteristics of these equipment, which are generally not measurable and time-variant during operation. Secondly, for better simulating the actual faults of electric power systems to promote test qualities, modern PRTDs are further required not only to be able to accurately output standard sinusoidal waveforms, but also to be able to accurately regenerate the fault waveforms recorded by digital fault recorders (DFRs) [1]. The fulfillment of these two requirements principally depends on the performance of power amplifiers (PAs) (including current PAs and voltage PAs) adopted by PRTDs.

Linear PAs (LPAs) [2,3] that consist of high-power transistors are prevalently deemed to have best adaptabilities in light of their ability to retain the linear relationships between command signals and output waveforms, and thus play as the main-stream in PRTD designs. The accuracies of most of the recent PRTDs adopting LPAs are claimed to be within $\pm 5\%$ for both current and voltage PAs. However, from detailed experiments, the authors found that the claimed accuracies are literally measured by the relative root mean square (RMS) errors of standard sinusoidal waveforms under fixed nominal loads, fixed system parameters and with no disturbances. For time-variant loads and time-variant system parameters, together with random disturbances, which are actually the general situations encountered in on-site testing, the errors are very likely to overreach the claimed range (sometimes overreach $\pm 5\%$) and tend to take a long time to recover. For fault waveforms that contain various frequency components and nonperiodic components, the accuracies degrade even more dramatically. These unfavorable problems directly influence the test qualities, and thereby most of PRTDs can merely meet the basic test demands.

In recent years, switching PAs (SPAs) [4,5] that are derived from power electronics technology are also applied in many waveform generation and amplification occasions. And compared with LPAs, SPAs have many advantages: (i) SPAs can spare digital-to-analog converters that are sometimes very expensive; (ii) the nominal capacity of a switching device is usually much higher than that of a high-power transistor, and thus for SPAs there is no need paralleling or cascading several devices to obtain a high output power; (iii) unlike LPAs, which need at least three complex amplifying stages to obtain a high amplifying gain, traditionally, SPAs need only one simple amplifying stage; (iv) the efficiency of SPAs is much higher than that of LPAs because the devices operate under a high-speed switching state; (v) in SPAs it is easier to isolate digital signals from high-power output signals by photoelectric couplers. On all accounts, the hardware structure of SPAs is much simpler than LPAs and is of higher efficiencies. If PRTDs adopt SPAs instead of LPAs, by virtue of the advantages of SPAs, the prices, volumes and weights of PRTDs can be better reduced, bringing a higher performance-price ratio.

However, the tracking accuracies of SPAs are more difficult to control than LPAs, which directly limits their applications in PRTDs, especially in modern PRTDs. Therefore, this paper is intended to design or find an effective control method to improve the tracking accuracies of SPAs to make their applications in modern PRTDs feasible. To begin with, the authors tested some modern control methods of power electronics technology. Repetitive control method [6–8], which is based on internal model principle, is a high-performance feedforward control strategy and can effectively track periodic signals and eliminate periodic disturbances or distortions; whereas, when command signals are nonperiodic or contain discontinuous periodic components, the dynamic response becomes slow and the tracking accuracy degrades significantly. Deadbeat control method [9,10], which is a superior predictive control strategy, has an excellent dynamic response and a good transient tracking ability; its tracking accuracy depends greatly on its predictive model, the choice of which is empirical and subjective, and thereby it is very difficult to choose a predictive model to ensure the adaptability to all the aforementioned situations. Sliding mode control method [11–13] shows a good robustness against system parameter variations once the operating point enters the predefined sliding surface; whereas, it is difficult to design an optimal sliding surface that can adapt to all types of the aforementioned situations; in addition, it is based on an ideal assumption that the sliding velocity of operating point is infinitely fast, which is unattainable in practical implementations due to the switching frequency limitations. In a word, these control methods are not satisfactorily suitable for the control of modern PRTDs.

It is in the process of testing and trying improving the control methods above that the authors discovered an interesting control method, which, to some extent, inherits both the characteristics of PID control method and deadbeat control method, and because it is more similar to the former it is named *quasi-PID control method*. Creative study shows that quasi-PID control method can be integrated with a single neuron, so that the self-adaptability to different loads, time-variant system parameters, random disturbances and fault waveforms containing various frequency components and nonperiodic components can be achieved conveniently and effectively, and this integrated control method is further named *single-neuron quasi-PID control method*. This paper is dedicated to discussing the derivation details of single-neuron quasi-PID control method and illustrating the satisfactory performance of SPAs brought by this control method and the corresponding applications in modern PRTDs.

MODELING OF SPAS

The SPAs discussed in this paper are based on a general purpose topology that consists of a single-phase full-bridge circuit and an independent DC source and can serve as both a current PA and a voltage PA (shown in Fig. 1). And several individual topologies of this kind can be combined to form multiple-channel outputs if needed. In Fig. 1, Q1–Q4 are insulated-gate bipolar transistors (IGBTs), D1–D4 are fast-recovery free-wheeling diodes, L and C are the inductor and capacitor of LC output filter, R is load resistor, \bar{V}_{dc} (assumed to be a constant) is the average voltage of DC

source, i_L is inductor current, i_R is load current, $v_R=i_R R$ is load voltage, and u_{mod} is modulation signal. The snubber circuits are omitted for clarity, the design of which is presented in [14]. Since the loads of PRTDs, nowadays, are mostly the microcomputer controlled protective relays and security and automatic equipment, the loads are actually the instrumental transformers ahead of the analog-to-digital converters of these equipment. The inductances of instrumental transformers are generally negligible (less than $1.0\mu\text{H}$), and hence they can be equivalent to sheer resistive loads of a few ohms (generally $3\sim 10\Omega$), which are signified by R in Fig. 1.

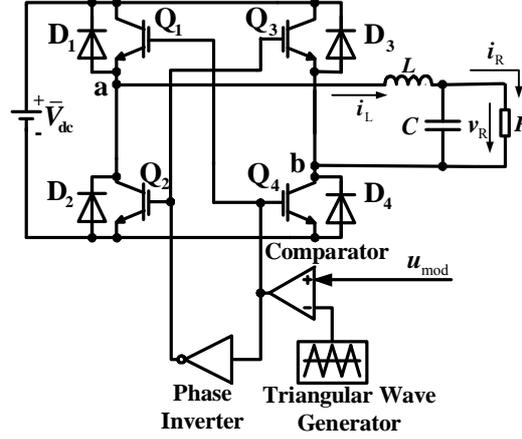


Figure 1. A general purpose topology of SPAs.

Open-loop model

Fig. 2 illustrates the principle of generating the control signals of SPAs – the bipolar pulse width modulation (PWM) signals, where V_c is the amplitude of isosceles triangle carrier, T_s is carrier period and also switching period and sampling period (sampling frequency $f_s=1/T_s$). According to the equivalent-area principle [15,16], the area of curved-edge trapezoidal pulse ABCDE should be equal to the net area of PWM pulses, i.e. $S_B=S_{B1}-S_{B2}-S_{B3}$. Because T_s is very small, the area of ABCDE is close to the area of rectangle A'C'DE (the shadowed area S_A). S_A and S_B can be written as

$$S_A = u_{\text{modc}} T_s \quad (1)$$

$$S_B = S_{B1} - S_{B2} - S_{B3} = \bar{V}_{\text{dc}} t_{\text{on}} - \bar{V}_{\text{dc}} t_{\text{off}} - \bar{V}_{\text{dc}} t_{\text{off}} \quad (2)$$

where u_{modc} is the ordinate of intersection point B, $t_{\text{on}} \in [0, T_s]$ and $t_{\text{off}} \in [0, T_s/2]$ are the turn-on time and turn-off time of Q1 and Q4. From $S_A = S_B$, the following relationship is obtained:

$$u_{\text{modc}} = (t_{\text{on}} - T_s/2) (2\bar{V}_{\text{dc}}/T_s) \quad (3)$$

where the relationship $t_{\text{on}} = T_s - 2t_{\text{off}}$ is considered. The prerequisite for accurate tracking is that the average output voltage \bar{v}_{ab} between points a and b (seen in Fig. 1) should be equal to u_{modc} , thus

$$\bar{v}_{\text{ab}} = u_{\text{modc}} = t_{\text{bon}} K_{\text{tv}}, \quad t_{\text{bon}} \in [-T_s/2, T_s/2] \quad (4)$$

where define $t_{\text{bon}} = t_{\text{on}} - T_s/2$ (s) and $K_{\text{tv}} = 2\bar{V}_{\text{dc}}/T_s$ (V/s). t_{bon} is called offset turn-on time (with an offset $-T_s/2$), and K_{tv} is called time-to-voltage transfer coefficient. It is easy to see that: (i) if $t_{\text{on}} = 0$ ($t_{\text{bon}} = -T_s/2$), then $\bar{v}_{\text{ab}} = -\bar{V}_{\text{dc}}$; (ii) if $t_{\text{on}} = T_s/2$ ($t_{\text{bon}} = 0$), then $\bar{v}_{\text{ab}} = 0$; (iii) if $t_{\text{on}} = T_s$ ($t_{\text{bon}} = T_s/2$), then $\bar{v}_{\text{ab}} = \bar{V}_{\text{dc}}$. Hence, (4) uniformly describes the turn-on process and turn-off process, and is literally the open-loop model of SPA.

Closed-loop model

Based on open-loop model, to realize closed-loop control the output of SPA should be fed back to affe-

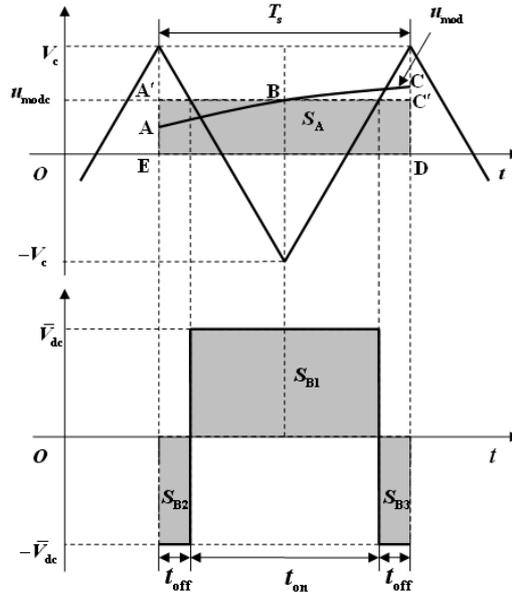


Figure 2. Principle of generating the bipolar PWM signals.

ct u_{mod} , and there are two types of ways: (i) let $u_{\text{mod}} = K_{ie} e_i = K_{ie} (i_R^* - i_R)$; (ii) let $u_{\text{mod}} = K_{ve} e_v = K_{ve} (v_R^* - v_R)$. i_R^* and v_R^* are current and voltage com-mand signals, e_i and e_v are output current and output voltage deviations, K_{ie} and K_{ve} are current error and voltage error amplification coefficients. By Way (i), current SPAs are realized, and by Way (ii), voltage SPAs are realized, whereas their circuits are the same. Given that the accuracy of current tracking is much more difficult to control than that of voltage tracking and that the control method of current SPAs can be transplanted to voltage SPAs by simply changing u_{mod} from $K_{ie} e_i$ to $K_{ve} e_v$, the following sections focus on analyzing current tracking only.

Continuous model in frequency domain

From the discussions of the foregoing two subsec-tions and considering also that the output of SPA needs at least one sampling period to be fed back to the controller, the continuous model of SPA in fre-quency domain can be constructed as shown in Fig. 3. The two blocks enclosed by dash-dotted frame are the *extended model* of SPA that combines SPA, *LC* output filter and the load together as a whole:

$$G(s) = \frac{I_R(s)}{T_{\text{bon}}(s)} = \frac{K_{iv} e^{-T_s s}}{RLCs^2 + sL + R} \quad (5)$$

where $I_R^*(s)$, $I_R(s)$ and $T_{\text{bon}}(s)$ are the Laplace trans-forms of i_R^* , i_R and t_{bon} , $D(s)$ is the transfer function of the controller, and F is the proportional coefficient of feedback channel. Though it is easy to write out the closed-loop transfer function according to Fig. 3, it is difficult to design the controller due to the pure-delay term $e^{-T_s s}$ which leads the system to be a non-minimum phase system [17]. And if the system is approximately turned into a minimum phase system by expanding $e^{-T_s s}$ into a power series and taking a finite number of the fore terms, the system would suffer from a great loss of bandwidth and thus induce high frequency component distortions. Hence, it is necessary to try to design the controller from a different angle, i.e., design the controller in time domain.

Discrete model in time domain

To design the controller in time domain, $G(s)$ must be discretized. The first step to do so is to transform $G(s)$ in s domain to $G(z)$ in z domain by virtue of the relation between Laplace transform and z transform:

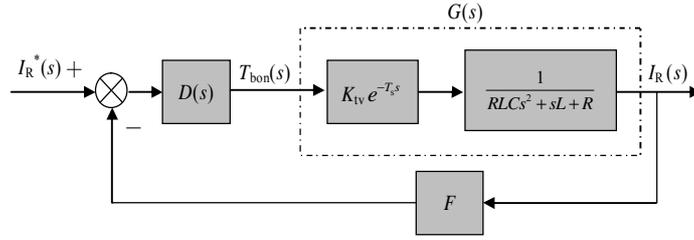


Figure 3. Continuous model of SPA in frequency domain.

$$G(z) = \left[\frac{1 - e^{-T_s s}}{s} G(s) \right] = \left[\frac{1 - e^{-T_s s}}{s} \frac{K_{tv} e^{-T_s s}}{RLCs^2 + sL + R} \right] \quad (6)$$

where $[\cdot]$ denotes performing z transform on the expressions in the square brackets. To maintain the invariability of the system step response after z trans-form, a zero-order holder $(1 - e^{-T_s s})/s$ is introduced to (6). The result of (6) is very complex if expressed with parameter symbols, therefore, instead, it is ex-pressed in numerical type with detailed values of the parameters substituted into the expressions and cal-culated (the values of the parameters are listed in Appendix A):

$$G(z) = \frac{I_R(z)}{T_{bon}(z)} = \frac{2.479 \times 10^4 z^{-2} + 1.845 \times 10^4 z^{-3}}{1 - 1.315 z^{-1} + 0.412 z^{-2}} \quad (7)$$

The second step is to perform inverse z transform on $G(z)$ to get the difference equation (the discrete model in time domain):

$$i_R(k) = 1.315 i_R(k-1) - 0.412 i_R(k-2) + 2.479 \times 10^4 t_{bon}(k-2) + 1.845 \times 10^4 t_{bon}(k-3) \quad (8)$$

where k is the integer index of discrete time series, $k=0, 1, 2, 3, \dots$. It is assumed: $i_R(-1) = i_R(-2) = 0$ and $t_{bon}(-1) = t_{bon}(-2) = t_{bon}(-3) = 0$.

QUASI-PID CONTROL METHOD

After the discrete model of SPA in time domain is derived, the discrete controller in time domain is to be designed in this section.

By reviewing Fig. 1, the Kirchoff voltage and current equations of SPA circuit can be written out:

$$(2p-1)\bar{V}_{dc} = L \frac{di_L}{dt} + i_L r + v_R \quad (9)$$

$$i_L = C \frac{dv_R}{dt} + i_R = RC \frac{di_R}{dt} + i_R \quad (10)$$

where $r = r_Q + r_L$ (r_Q is the equivalent switching resi-stance of IGBT, r_L is the winding resistance of L), p is a unipolar two-valued-logic switching function:

$$p = \begin{cases} 1 & \begin{pmatrix} Q_1 \text{ and } Q_4 \text{ ON or } D_1 \text{ and } D_4 \text{ ON} \\ Q_2 \text{ and } Q_3 \text{ OFF and } D_2 \text{ and } D_3 \text{ OFF} \end{pmatrix} \\ 0 & \begin{pmatrix} Q_2 \text{ and } Q_3 \text{ ON or } D_2 \text{ and } D_3 \text{ ON} \\ Q_1 \text{ and } Q_4 \text{ OFF and } D_1 \text{ and } D_4 \text{ OFF} \end{pmatrix} \end{cases} \quad (11)$$

where ON and OFF denote “turn on” and “turn off”.

When symmetric regular sampling is adopted in the modulating process as shown in Fig. 2, it is easy to write out the duty cycle

$$D = \frac{t_{on}}{T_s} = \frac{u_{modc} + V_c}{2V_c} \approx \frac{K_{ie} e_i}{2V_c} + \frac{1}{2} \quad (12)$$

where the duration time for $p=1$ is t_{on} and that for $p=0$ is t_{off} .

Given that T_s is very small, the integration of current differential di_L within a T_s is equal to the summation of small current variations, which is approximate to inductor current variation Δi_L . Thus, by integrating both sides of (9) over a T_s , an equation is obtained:

$$\begin{aligned}\Delta i_L &\approx \int_0^{T_s} di_L = \frac{1}{L} \int_0^{T_s} [(2p-1)\bar{V}_{dc} - (i_L r + v_R)] dt \\ &= \frac{1}{L} \left\{ 2[-\bar{V}_{dc} - (i_L r + v_R)]t_{\text{off}} + [\bar{V}_{dc} - (i_L r + v_R)]t_{\text{on}} \right\} \\ &= \frac{T_s}{L} [(2D-1)\bar{V}_{dc} - (i_L r + i_R R)] \\ &= \frac{T_s}{L} \left[\frac{K_{ie} e_i \bar{V}_{dc}}{V_c} - (i_L r + i_R R) \right] \quad (13)\end{aligned}$$

When i_L increases, i.e., $\Delta i_L \geq 0$, from (13), it equates

$$e_i \geq \frac{V_c (i_L r + i_R R)}{K_{ie} \bar{V}_{dc}} = e_i^* \neq 0 \quad (14)$$

where e_i^* is defined and called *inherent tracking error*. Likewise, when i_L decreases, i.e., $\Delta i_L \leq 0$, (14) becomes $e_i \leq e_i^* \neq 0$. Here, $e_i^* \neq 0$ indicates that e_i is fluctuating around a nonzero value, meaning that the error exists all the time and so it is inherent. That is to say, merely using $u_{\text{mod}} = K_{ie} e_i$, the closed-loop system can not realize non-static-error tracking and the attainment of high tracking accuracy is impossible.

In order to counteract e_i^* , the authors creatively construct a *modified current command signal*:

$$\hat{i}_R^* = i_R^* + e_i^* = i_R^* + \frac{V_c (i_L r + i_R R)}{K_{ie} \bar{V}_{dc}} \quad (15)$$

By replacing the i_R^* in e_i ($e_i = i_R^* - i_R$) in (12) and (13) with \hat{i}_R^* , the modified duty cycle \hat{D} and the modified inductor current variation $\Delta \hat{i}_L$ are written as

$$\hat{D} = \frac{K_{ie} (\hat{i}_R^* - i_R)}{2V_c} + \frac{1}{2} = \frac{K_{ie} e_i}{2V_c} + \frac{i_L r + i_R R}{2\bar{V}_{dc}} + \frac{1}{2} \quad (16)$$

$$\Delta \hat{i}_L \approx \frac{T_s}{L} \left[\frac{K_{ie} (\hat{i}_R^* - i_R) \bar{V}_{dc}}{V_c} - (i_L r + i_R R) \right] = \frac{e_i T_s K_{ie} \bar{V}_{dc}}{LV_c} \quad (17)$$

From (17), it is seen that whether $\Delta \hat{i}_L$ increases or decreases, e_i is fluctuating around 0 now, meaning that e_i^* is eliminated.

The coefficient of e_i in (17) can be forced to be equal to 1, then

$$K_{ie} = \frac{LV_c}{T_s \bar{V}_{dc}} = \frac{LV_c f_s}{\bar{V}_{dc}} \quad (18)$$

and this leads to a concise form of (17):

$$\Delta \hat{i}_L \approx e_i \quad (19)$$

Hereto, the duty cycle of (16) is the continuous control quantity (continuous controller) of SPA in time domain, and so it is needed to be discretized to obtain its discrete type in time domain:

$$\hat{D}(k) = \frac{K_{ie} e_i(k)}{2V_c} + \frac{i_L(k)r + i_R(k)R}{2\bar{V}_{dc}} + \frac{1}{2} \quad (20)$$

and the incremental type $\Delta \hat{D}(k) = \hat{D}(k) - \hat{D}(k-1)$ is

$$\begin{aligned}\Delta \hat{D}(k) &= \frac{K_{ie}}{2V_c} [e_i(k) - e_i(k-1)] + \frac{r}{2\bar{V}_{dc}} [i_L(k) - i_L(k-1)] + \\ &\quad \frac{R}{2\bar{V}_{dc}} [i_R(k) - i_R(k-1)] \quad (21)\end{aligned}$$

Similarly, (10) is also needed to be discretized:

$$i_L(k) = \frac{RC}{T_s} [i_R(k) - i_R(k-1)] + i_R(k) \quad (22)$$

where the first-order backward difference is adopted to approximate the first-order differential. And the incremental type of (22) $\Delta i_L(k) = i_L(k) - i_L(k-1)$ is

$$\Delta i_L(k) = \frac{RC}{T_s} [i_R(k) - 2i_R(k-1) + i_R(k-2)] +$$

$$i_R(k) - i_R(k-1) \quad (23)$$

Besides, the discretized type of (19) is

$$\Delta \hat{i}_L(k) = i_L(k) - i_L(k-1) \approx e_i(k) \quad (24)$$

Substituting (24) into (23), (23) can be rearranged as

$$i_R(k) - i_R(k-1) =$$

$$e_i(k) - \frac{RC}{T_s} [i_R(k) - 2i_R(k-1) + i_R(k-2)] \quad (25)$$

Then substituting (24) and (25) into (21), (21) becomes

$$\Delta \hat{D}(k) = \frac{K_{ie}}{2V_c} [e_i(k) - e_i(k-1)] + \frac{r+R}{2\bar{V}_{dc}} e_i(k) -$$

$$\frac{R^2C}{2\bar{V}_{dc}T_s} [i_R(k) - 2i_R(k-1) + i_R(k-2)] \quad (26)$$

(26) is the required discrete controller of SPA in time domain.

On the other hand, a widely used type of PID control method [17] is

$$u(k) = K_p e(k) + K_I \sum_{j=0}^k e(j) T_s + \frac{K_D}{T_s} [e(k) - e(k-1)] \quad (27)$$

where $u(k)$ is control quantity, $e(k)$ is the error between output and command signal, K_p , K_I and K_D are P, I and D parameters. The incremental type of (27) $\Delta u(k) = u(k) - u(k-1)$ is

$$\Delta u(k) = K_p [e(k) - e(k-1)] + K_I T_s e(k) +$$

$$\frac{K_D}{T_s} [e(k) - 2e(k-1) + e(k-2)] \quad (28)$$

A term-to-term comparisons of (26) and (28) reveal that the first two terms of both equations are in accordance with each other, whereas the third term of (26) is composed of $i_R(k)$ and that of (28) is composed of $e(k)$. Therefore, (28) is not a real PID controller, yet it does have a structure similar to that of a PID controller. Due to this, (28) is named *quasi-PID control method*. From the comparisons, it is easy to write out the quasi-PID parameters:

$$K_p = \frac{K_{ie}}{2V_c} = \frac{Lf_s}{2\bar{V}_{dc}}, K_I = \frac{(r+R)f_s}{2\bar{V}_{dc}}, \hat{K}_D = \frac{-R^2C}{2\bar{V}_{dc}} \quad (29)$$

where the quasi-D parameter is denoted as \hat{K}_D to be distinguished from K_D . Based on (29), (26) is simplified as

$$\Delta \hat{D}(k) = K_p [e_i(k) - e_i(k-1)] + K_I T_s e_i(k) +$$

$$\frac{\hat{K}_D}{T_s} [i_R(k) - 2i_R(k-1) + i_R(k-2)] \quad (30)$$

Discussion: From (30), it should be noted again that the quasi-D term is not a real D term, and if a real D term is added into (30) to turn it into a real PID control method, due to the lack of the theoretical support, it will not be able to counteract the inherent tracking error but inversely detrimentally influence the system stability like an extra and undesirable disturbance. In a word, the strict PID

control method is not suitable for SPA or inverter control and this is the reason why few references reported such an application.

Finally, considering that the actual control quantity in (8) is $t_{\text{bon}}(k)$, $\Delta\hat{D}(k)$ should be converted to $t_{\text{bon}}(k)$ for real application:

$$\begin{aligned}\Delta t_{\text{bon}}(k) &= t_{\text{bon}}(k) - t_{\text{bon}}(k-1) = \Delta t_{\text{on}}(k) = T_s \Delta\hat{D}(k) \\ &= K_p T_s [e_i(k) - e_i(k-1)] + K_i T_s^2 e_i(k) + \\ &\quad \hat{K}_D [i_R(k) - 2i_R(k-1) + i_R(k-2)]\end{aligned}\quad (31)$$

The ability to counteract the inherent tracking error is illustrated in Subsection 5.1.

SINGLE-NEURON QUASI-PID CONTROL METHOD

Although quasi-PID control method can counteract the inherent tracking error, it is still not perfect. Equation (29) shows that the three quasi-PID parameters are closely correlated with circuit parameters L , f_s , \bar{V}_{dc} , r , R and C . These seemingly constant parameters actually vary with time and operating conditions, for example: (i) the fluctuation of output power may lead to the fluctuation of \bar{V}_{dc} , so the pre-sumption that \bar{V}_{dc} is a constant is discounted; (ii) R is different with different tested equipment and can-not be accurately measured, and is always drifting with temperature; (iii) the nonlinear variations of r_Q and r_L may make r ripple nonlinearly; and so on. These may influence the accuracy of quasi-PID parameters and induce tracking errors. In addition, the dead-time embedded in turn-on time, the side effect of snubber circuits and other random disturbances may introduce extra errors. To further eliminate all these errors, on trial, the authors accidentally found that quasi-PID control method can be integrated with a single neuron harmoniously, and so by virtue of the adaptability of the single neuron the foregoing errors can be adaptively and rapidly compensated online. The integrated control method is named *single-neuron quasi-PID control method*.

Adaptive control structure

The structure of single-neuron quasi-PID control method is presented in Fig. 4, where $x_1(k)$, $x_2(k)$ and $x_3(k)$ are the three inputs of single neuron:

$$\begin{cases} x_1(k) = e_i(k) - e_i(k-1) \\ x_2(k) = e_i(k) = i_R^*(k) - i_R(k) \\ x_3(k) = i_R(k) - 2i_R(k-1) + i_R(k-2) \end{cases}\quad (32)$$

And $w_1(k)$, $w_2(k)$ and $w_3(k)$ are three connection weights:

$$w_1(k) = K_p T_s, \quad w_2(k) = K_i T_s^2, \quad w_3(k) = \hat{K}_D\quad (33)$$

The single neuron sums three weighted inputs up by adder component “ Σ ” to form a total input signal:

$$S_{\text{in}}(k) = \sum_{j=1}^3 w_j(k) x_j(k) = \Delta t_{\text{bon}}(k)\quad (34)$$

Substituting (32) and (33) into (34), it is seen that (34) realizes the very same calculation of (31).

The three connection weights should be normalized to maintain their relative magnitudes to promote the robustness of simulation and actual control. The normalization can be realized by vector norms. There are three commonly used vector norms [18]: (i) 1-norm, the summation of the absolute values of the elements; (ii) 2-norm, the square root of the quadratic sum of the elements; (iii) ∞ -norm, the maximum value of the absolute values of the elements. Comparisons show that 2-norm is of the greatest computational complexity, and simulations show that it does not give a better control effect than 1-norm. Although ∞ -norm is of the lowest computational complexity, it always makes one of the three normalized connection weights equal to 1, causing the corresponding input to have the greatest impact on control quantity and thus inducing oscillations in output waveforms during the first 1 or 2 periods. Hence, 1-norm is the best choice, and the normalized type of (34) based on 1-norm is

$$S_{\text{in(norm)}}(k) = \sum_{j=1}^3 \frac{w_j(k)}{\|\mathbf{w}\|_1} x_j(k) = \sum_{j=1}^3 \frac{w_j(k)}{\sum_{n=1}^3 |w_n(k)|} x_j(k) = \sum_{j=1}^3 \bar{w}_j x_j(k) \quad (35)$$

where $\mathbf{w} = [w_1(k), w_2(k), w_3(k)]$ is connection weight vector, $\|\mathbf{w}\|_1$ is the 1-norm of \mathbf{w} , and \bar{w}_j is defined to replace the coefficient of $x_j(k)$ for short.

The single neuron then takes $S_{\text{in(norm)}}(k)$ through its excitation function $f(\cdot)$ to generate the normalized control quantity:

$$\Delta t_{\text{bon(norm)}}(k) = f[S_{\text{in(norm)}}(k)]$$

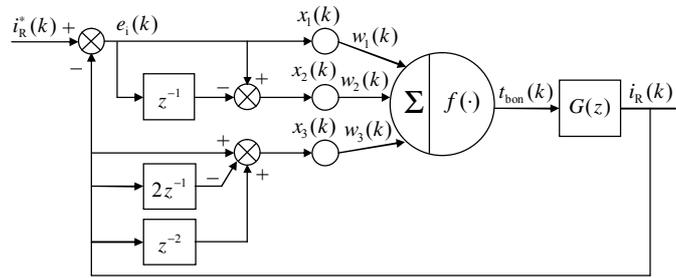


Figure 4. Structure of single-neuron quasi-PID control method.

$$= \begin{cases} 5 & S_{\text{in(norm)}}(k) \in (5, \infty) \\ K_{\text{sl}} S_{\text{in(norm)}}(k) & S_{\text{in(norm)}}(k) \in [-5, 5] \\ -5 & S_{\text{in(norm)}}(k) \in (-\infty, -5) \end{cases} \quad (36)$$

where a linear proportional function with amplitude limitations is chosen as $f(\cdot)$, and K_{sl} is the slope of the linear segment of $f(\cdot)$. The choice of this excitation function lies on two considerations: (i) the amplitude limitation of the control quantity is indispensable to prevent the control quantity from over-reaching; (ii) $S_{\text{in(norm)}}(k)$ has already been the required control quantity, where further process with complex excitation function, such as sigmoid function or radial basis function, may not only deprive its physical meanings but induce unnecessary computational complexities, so it is better to choose a simpler function. Next, it should be noted that the amplitude limitations of $f(\cdot)$ are set as ± 5 instead of ± 1 (± 1 are the classic amplitude limitations defined in normalization theory), and the reasons are: (i) to avoid pure decimal-fraction computations on fixed point microprocessor (used in this paper), as may introduce large rounding errors; (ii) to slightly loosen the amplitude limitations to enhance the fault tolerance of the control method.

In (8), the coefficients of $t_{\text{bon}}(k-2)$ and $t_{\text{bon}}(k-3)$ are far greater than those of $i_{\text{R}}(k-1)$ and $i_{\text{R}}(k-2)$ because the variables and their coefficients are all actual values. Seeing that (35) and (36) are in normalized types, (8) cannot be calculated with them if not normalized accordingly. Two steps are taken to normalize $t_{\text{bon}}(k)$ and $i_{\text{R}}(k)$: (i) according to (4) and (36), where $t_{\text{bon}} \in [-T_s/2, T_s/2]$ and $t_{\text{bon(norm)}}(k) \in [-5, 5]$, when $T_s = 1 \times 10^{-4}$ s, $t_{\text{bon}}(k) \in [-5 \times 10^{-5}$ s, 5×10^{-5} s], so $t_{\text{bon}}(k) = t_{\text{bon(norm)}}(k) \times 10^{-5}$; (ii) the peak-to-peak values of the output current in this paper are designated as ± 10 A, so $i_{\text{R}}(k)$ is normalized by being divided by 10 A, and the result is the per-unit value $i_{\text{R(norm)}}(k) = i_{\text{R}}(k)/10$. As a result, (8) is normalized as

$$i_{\text{R(norm)}}(k) = 13.15 i_{\text{R(norm)}}(k-1) - 4.12 i_{\text{R(norm)}}(k-2) + 0.2479 t_{\text{bon(norm)}}(k-2) + 0.1845 t_{\text{bon(norm)}}(k-3) \quad (37)$$

Adaptive learning algorithm

The adaptability of single neuron is realized by adaptive adjustment of connection weights. The general learning rule [19] for connection weight adjustment is as follows:

$$\mathbf{w} = \eta g(\mathbf{w}, \mathbf{x}, d) \mathbf{x} - \lambda \mathbf{w} \quad (38)$$

where \mathbf{w} is connection weight vector, $\Delta\mathbf{w}$ is the incremental vector of \mathbf{w} , $\eta > 0$ is learning rate, \mathbf{x} is input vector, d (a scalar quantity) is the expected output and is called *teacher signal*, function $g(\cdot)$ is learning signal and $\lambda \geq 0$ is a real constant.

Discussion 1: If $g(\mathbf{w}, \mathbf{x}, d) = d - y = e$ (e is the error between d and the actual output y , e and y are scalar quantities) and $\lambda = 0$, then $\Delta\mathbf{w} = \eta(d - y)\mathbf{x}$, which is *perceptron learning rule* based on least mean square standard. This learning rule includes d , hence it is a supervised learning rule with teacher guidance, and theory [20] verifies that it is asymptotically stable. As to this learning rule, (38) is expanded as

$$w_j(k) = \eta_j [d(k) - y(k)] x_j(k) = \eta_j e(k) x_j(k) \quad (39)$$

where, in this paper, $d(k) = i_{R(\text{norm})}^*(k)$, $y(k) = i_{R(\text{norm})}(k)$, $e(k) = e_i(k)$ and $j = 1, 2, 3$. Simulations show that this learning rule possesses outstanding stability but lacks independence or self-learning enthusiasm. When illustrated on output waveforms, the phenomenon is that the steady-state errors of output waveforms are very small while the response speed is fairly low.

Discussion 2: If $g(\mathbf{w}, \mathbf{x}, d) = u$ (u is control quantity and is a scalar quantity) and $\lambda = 0$, then $\Delta\mathbf{w} = \eta u \mathbf{x}$, which is *Hebb learning rule*. This learning rule does not include d , hence it is an unsupervised learning rule without teacher guidance, and theory [20] verifies that it is unstable under certain conditions. As to this learning rule, (38) is expanded as

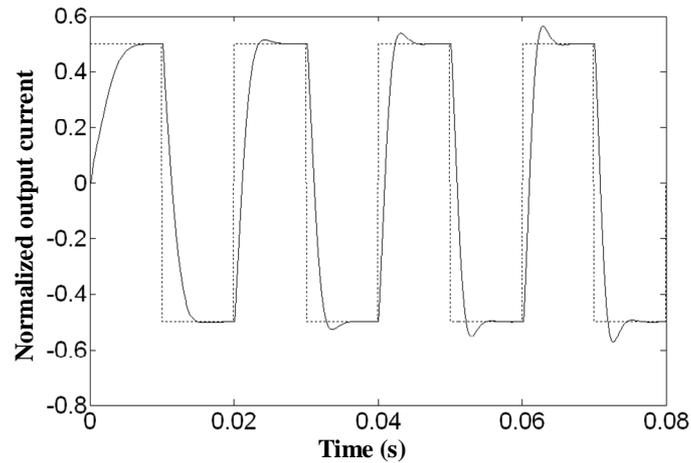
$$w_j(k) = w_j(k) - w_j(k-1) = \eta_j u(k) x_j(k) \quad (40)$$

where, in this paper, $u(k) = t_{\text{bon}(\text{norm})}(k)$. Simulations show that this learning rule has strong independence and self-learning ability, and its learning speed is very fast. When illustrated on output waveforms, the phenomenon is that the response speed is fairly high, while, due to the lack of teacher guidance, the steady-state errors are relatively large.

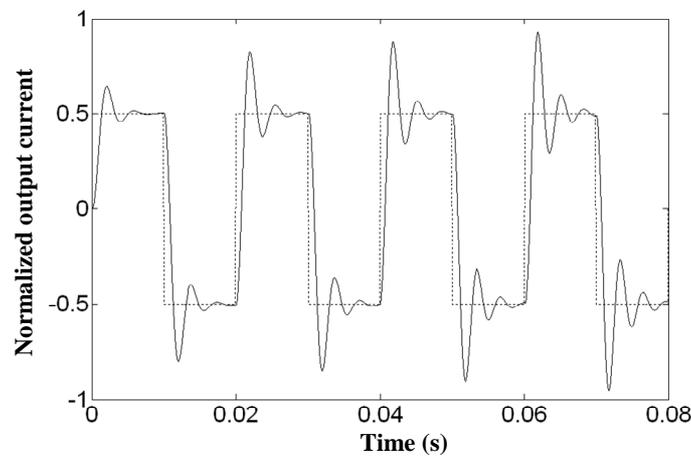
To better illustrate the two discussions above, a periodic square waveform is chosen as a simulation example. The reasons for the choice are: (i) for periodic waveform, comparisons can be made between different waveforms or among different segments of the same waveform; (ii) for square waveform, it has rising and falling edges and smooth segments, so the steepness of the former can be used to assess response speed and the smoothness of the latter can be used to assess steady-state errors. The simulated waveform using perceptron learning rule is presented in Fig. 5(a), showing that the rising and falling edges are not steep (implying that the response speed is low) but the smooth segments are very flat (implying that the steady-state errors are very small). The simulated waveform using Hebb learning rule is presented in Fig. 5(b), showing that the rising and falling edges are much steeper than those in Fig. 5(a) (implying that the response speed is high) but there exist oscillations and great overshoots in smooth segments (implying that the steady-state errors are large), and the oscillations seem to grow larger and larger, indicating the likelihood to become unstable.

Given that the strong point of perceptron learning rule is the weak point of Hebb learning rule and vice versa, the authors creatively combine them together and propose the *perceptron-Hebb learning rule*, the expanded type of which is

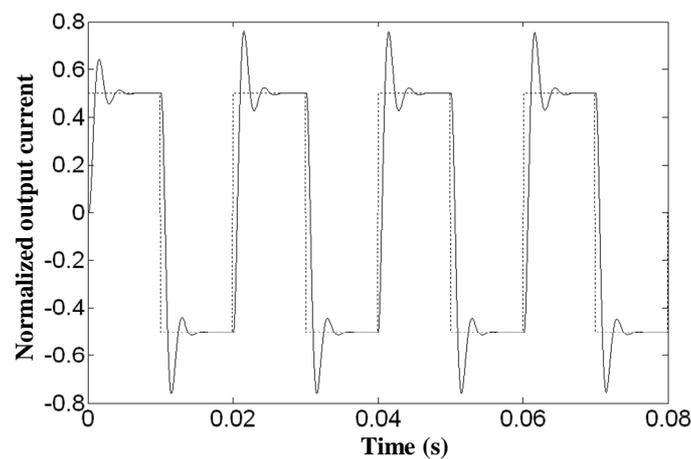
$$w_j(k) = w_j(k) - w_j(k-1) = \eta_j e(k) u(k) x_j(k) \quad (41)$$



(a) Perceptron learning rule.



(b) Hebb learning rule.



(c) Perceptron-Hebb learning rule.

Figure 5. Simulated periodic square waveforms using three learning rules. Dotted: command signal; solid: simulated waveform.

The simulated waveform using this new learning rule is presented in Fig. 5(c), which shows that the rising and falling edges are steeper than those in Fig. 5(a) and the smooth segments are flatter than those in Fig. 5(b), implying that the low response speed of perceptron learning rule, and the large steady-state errors and possible instability of Hebb learning rule are integrally improved. In a word, perceptron-Hebb learning rule inherits both the strong points of perceptron learning rule and Hebb learning rule but gets rid of their weak points to a large extent.

Control flow and stability analysis

To sum up, the control flow of single-neuron quasi-PID control method is presented in Fig. 6. In the flow chart, (32), (41) and (36) are the three most complex computational procedures, but they demand only a small amount of floating additions and multi-plications. These calculations are of relatively low computational complexities, implying that the control method is very suitable for real-time control and for realization on low-performance microprocessors.

From (33), it is seen that $w_j(0)$ are actually initialized by quasi-PID parameters, which, as mentioned above, have small errors and will vary during operation. Hence, it is necessary to analyze the impacts of the inaccuracy of the initial $w_j(0)$ on system stability.

By performing z transforms on (36) and (37), then solving the resultant simultaneous equations, system function $H(z) = I_{R(\text{norm})}(z) / I_{R(\text{norm})}^*(z)$ can be obtained. After rationalizations of both the numerator and denominator polynomials of $H(z)$, the denominator polynomial becomes the characteristic polynomial that is in the following form:

$$F_{\text{ch}}(z) = a_5 z^5 + a_4 z^4 + a_3 z^3 + a_2 z^2 + a_1 z + a_0 \quad (42)$$

where a_0, a_1, \dots, a_5 are coefficients acquired from rationalizations and are not detailedly written out for clarity. In terms of Jury criteria [21,22], the constraint conditions for system stability are:

- (i) The first criterion requires $F_{\text{ch}}(1) > 0$, and it is fulfilled straightforwardly because a_0, a_1, \dots, a_5 are all positive numbers;
- (ii) The order of $F_{\text{ch}}(z)$ is 5 (an odd number), so the second criterion requires $F_{\text{ch}}(-1) < 0$, which gives the inequation

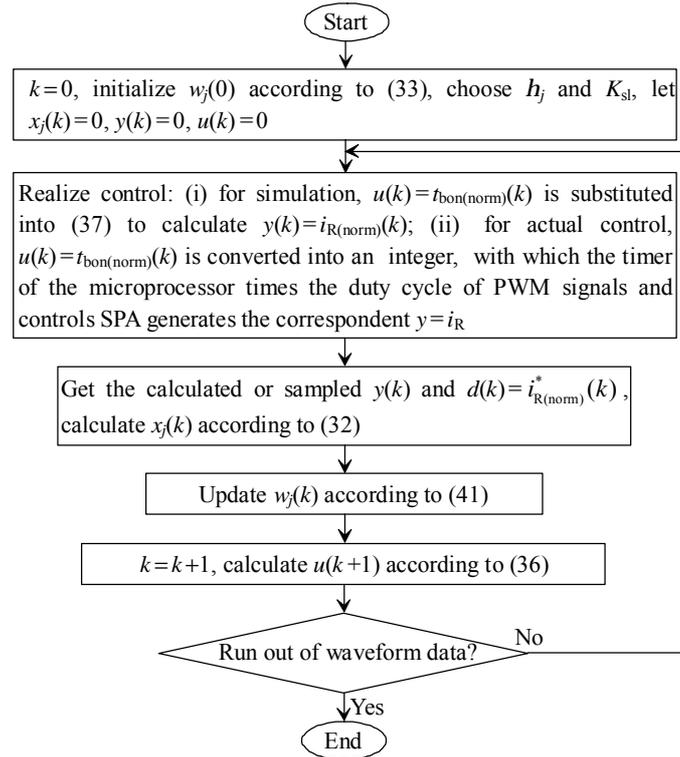


Figure 6. Principle of generating the bipolar PWM signals.

$$-0.00634\bar{w}_1 - 0.1268\bar{w}_2 + 0.2536\bar{w}_3 < 2.3454/K_{sl} \quad (43)$$

- (iii) The third criterion requires $|a_0| < a_5$, which gives the inequation

$$0.1845K_{sl}\bar{w}_3 < 1 \quad (44)$$

- (iv) ...

From the calculations above, it is found that as long as K_{sl} fulfills (43) and (44) at the same time, the curtailed inequations from (iv) are fulfilled as well; and that all the inequations have certain margins to retain their inequalities, which not only gives the choice of K_{sl} certain freedom, but also makes the impacts of the inaccuracy of the initial $w_j(0)$ negligible. In short, as long as $w_j(0)$ are initialized according to (33) and K_{sl} is selected according to (43) and (44), system stability is ensured.

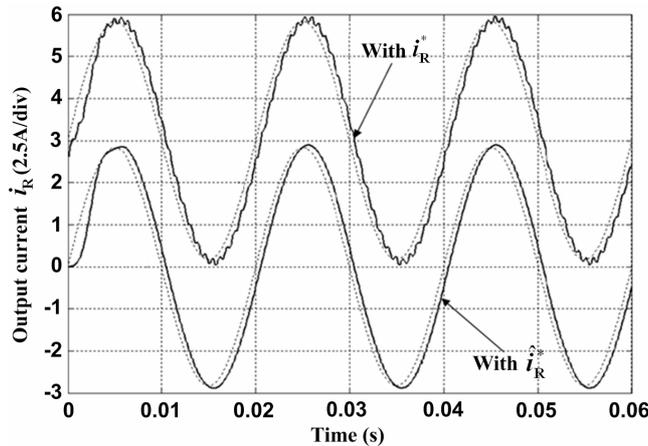
SIMULATION AND EXPERIMENTAL RESULTS

In this section, the effectiveness of single-neuron quasi-PID control method is illustrated by four groups of simulation and experimental results. The experiments are implemented on a prototype machine, which is shown in Appendix B. Subsection 5.1 illustrates the sheer ability of quasi-PID control method (without single neuron) to counteract the inherent tracking error. The next three subsections illustrate the adaptabilities of single-neuron quasi-PID control method to different loads, time-variant system parameters, random disturbances and fault waveforms containing various frequency components and nonperiodic components. And at the end, the accuracies of output fault current waveforms generated by various control methods are compared.

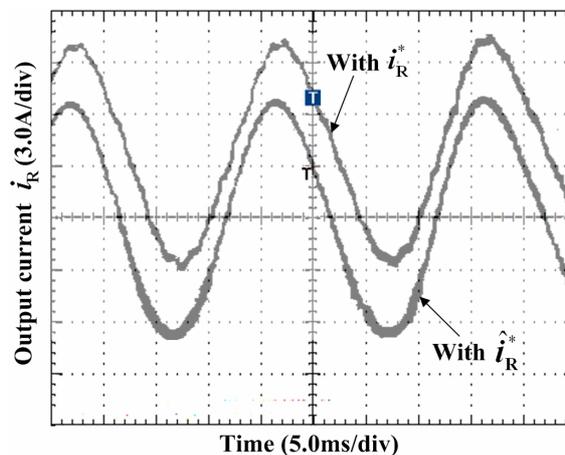
Ability of quasi-PID control method to counteract inherent tracking error

A 5 A (RMS), 50 Hz sinusoidal waveform is chosen for illustration, and the simulated and experimental waveforms of SPA output current respectively based on current command signal i_R^* and modified current command signal \hat{i}_R^* are presented in Fig. 7. Both Figs. 7(a) and (b) show that the output waveform based on i_R^* fluctuates around the command signal and is distorted in the vicinity of the peaks, manifesting the existence and detrimental effects of the inherent tracking error; whereas, the output waveform based on \hat{i}_R^* is much smoother and has no distortions in the vicinity of the peaks, implying the effective elimination of the inherent tracking error.

From further observation of Fig. 7(b) and comparison with Fig. 10(f) in Subsection 5.4, it can be seen that the experimental waveform based on \hat{i}_R^* in Fig. 7(b) is a little thicker than that in Fig. 10(f), manifesting that without the aid of single neuron, quasi-PID control method is unable to dynamically adapt to the variations of loads and system parameters and random disturbances, and thus is unable to achieve



(a) Simulated waveforms. Dotted: command signals; solid: output waveforms.

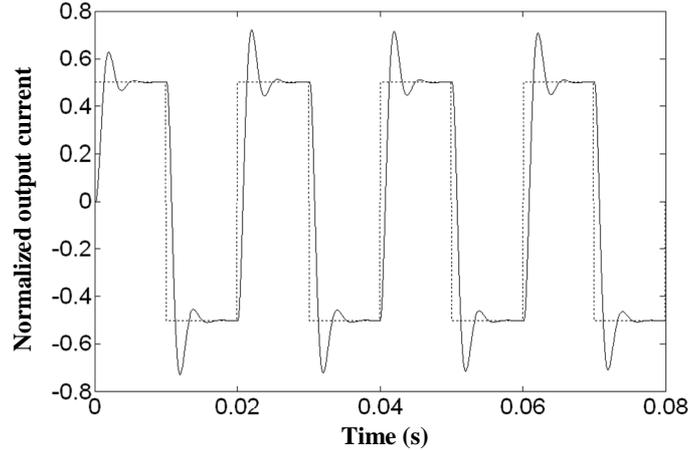


(b) Experimental waveforms.

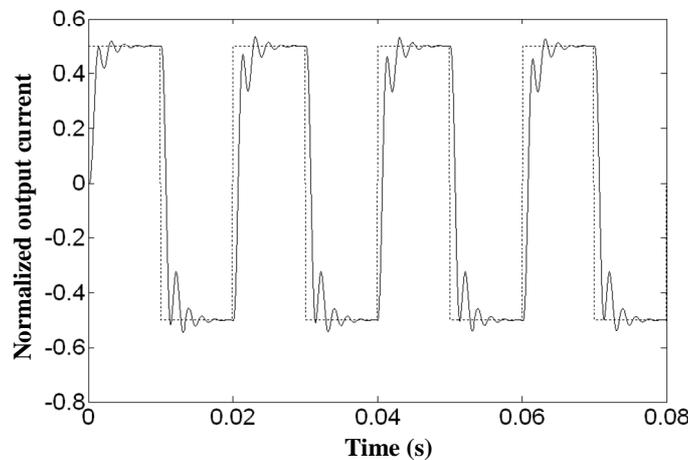
Figure 7. Ability of quasi-PID control method to counteract inherent tracking error. the required high accuracy.

Adaptability of single-neuron quasi-PID control method to different loads

A ± 5 A (peak-to-peak value), 50 Hz square wave-form (its characteristics were discussed in Subsection 4.2) is chosen for illustration. Two situations – $R=3\ \Omega$ and $10\ \Omega$ (the general lower limit and upper limit of R) are taken for examples, and the simulated waveforms are presented in Fig. 8. It is seen that except the overshoots the rising and falling edges of both Figs. 8(a) and (b) are alike, and the adjusting time and the smoothness of the smooth segments are also similar, manifesting that to different loads single-neuron quasi-PID control method adaptively adjusts to maintain the same response speed and steady-state errors. As to the difference of the over-shoots, the explanation is that to perfectly regenerate the rising and falling edges of a square waveform is something that is too ideal to reach for all control methods. Considering the similarity with the simulated waveforms, the experimental waveforms are curtailed for brevity.



(a) $R = 3\ \Omega$.



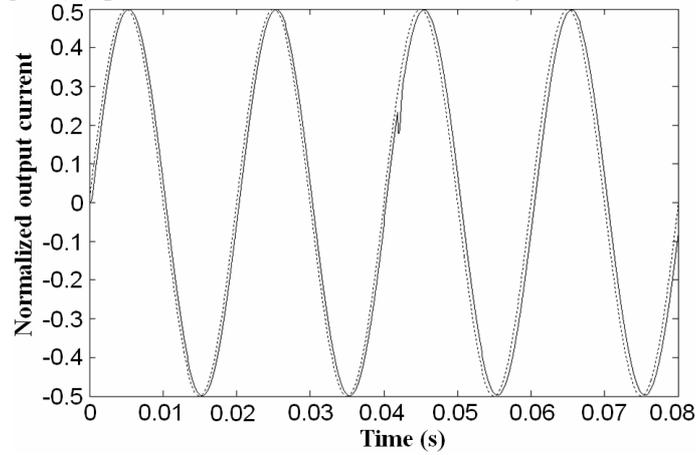
(b) $R = 10\ \Omega$.

Figure 8. Adaptability of single-neuron quasi-PID control method to different loads. Dotted: command signal; solid: simulated output waveform.

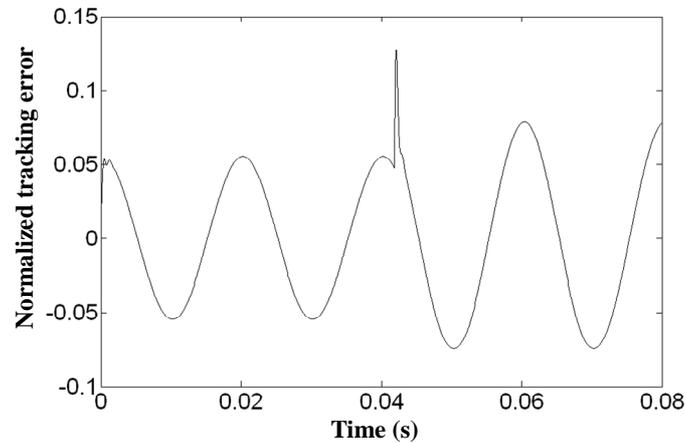
Adaptability of single-neuron quasi-PID control method to time-variant system parameters

As mentioned in Section 4, there are many types of time variations of system parameters. Considering the representativeness of the load resistor drift at different temperature, it is chosen as an illustrative example. Here, assume that R varies from $3\ \Omega$ to $5\ \Omega$ as temperature increases, and that the variation completes during $0.042\ \text{s} \sim 0.045\ \text{s}$. In fact, the variation is very slow in real world, based on which the correspondent variation may be too subtle to be observed on output waveform, hence the variation of R is artificially assumed to be fast, which encompasses the slow variation as its special case. A 5 A (RMS), 50 Hz sinusoidal waveform is chosen for illustration, and the simulated waveforms are presented in Fig. 9. From Fig. 9(a), it is seen that with the rapid self-adjustment of single-neuron quasi-PID control method the variation of R only induces a

small notch on the waveform and very soon the wave-form recovers smoothness and stability, from which it is almost impossible to sense that R has varied. To better illustrate the dynamic adjustment process, Fig. 9(b) presents the waveform of the dy-



(a) Dotted line: command signal; solid line: simulated output waveform.



(b) Dynamic tracking error.

Figure 9. Adaptability of single-neuron quasi-PID control method to time-variant system parameters.

dynamic tracking error e_i , showing that when R varies e_i varies accordingly and serves as a direct information source for the single neuron to make a change. Like Subsection 5.3, the experimental waveforms are curtailed for brevity.

Adaptability of single-neuron quasi-PID control method to different waveforms

In this subsection, different types of waveforms are chosen to illustrate the adaptability of single-neuron quasi-PID control method to periodic and nonperiodic waveforms, and to waveforms with singular or complex frequency components. The simulated and experimental results are presented in Figs. 10 and 11. It is seen that in Fig. 10 the simulated waveforms and the experimental waveforms are alike, and in Fig. 11 the experimental fault waveforms match the corresponding waveforms recorded by DFRs satisfactorily, meaning that from perceptual impression single-neuron quasi-PID control method can adapt to all kinds of waveforms.

A comparison of Fig. 10(f) and Fig. 7(b) (in Subsection 5.1) shows that the sinusoidal waveform in Fig. 10(f) is much slimmer and smoother than the sinusoidal waveform based on \hat{i}_R^* in Fig. 7(b), demonstrating that single-neuron quasi-PID control method can adaptively and effectively inhibit system parameters drift and random disturbances, and thus achieve a higher accuracy.

The foregoing arguments are totally derived from subjective impressions, which are superficial. Therefore, a numerical criterion is constructed for quantitatively assessing the accuracy of the

output waveforms, which is able to make point-to-point comparisons between the actual output waveform and the command signal. The mean square error (MSE), i.e. $MSE = E[(i_R^* - i_R)^2]$, is chosen as the numerical criterion, where the notation $E[\cdot]$ signifies the mathematical expectation computation. For discrete time series, $E[\cdot]$ should be estimated by a statistical quantity:

$$\hat{e}_{MSE} \approx \frac{1}{N} \sum_{k=1}^N [i_{R(norm)}^*(k) - i_{R(norm)}(k)]^2 \times 100\% \quad (45)$$

where N is the length of the time series.

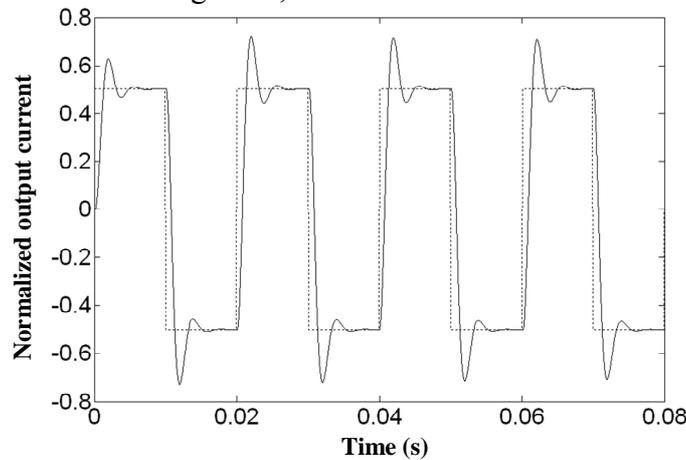
As an example, the fault current waveform recorded by DFR as shown in Fig. 11(a) is respectively generated by single-neuron quasi-PID control method, quasi-PID control method, PI control method, repetitive control method and deadbeat control method, and (45) is respectively performed on the sampling data of the resultant experimental fault current waveforms (the experimental fault current waveform generated by single-neuron quasi-PID control method is shown in Fig. 11(b), and others are curtailed for brevity), the results of which are presented in Table 1. From comparisons, the following conclusions can be drawn: (i) excluding single-neuron quasi-PID control method, deadbeat control method gives the best accuracy, but by eliminating the inherent tracking error, the accuracy of quasi-PID control method is comparable to deadbeat control method; (ii) the accuracy improvement owing to the single neuron is impressive, and this high accuracy of single-neuron quasi-PID control method not only makes the applications of SPAs in PRTDs feasible, but also makes the fault waveform regeneration on modern PRTDs more effective.

Table 1. MSEs of waveforms generated by 6 control methods.

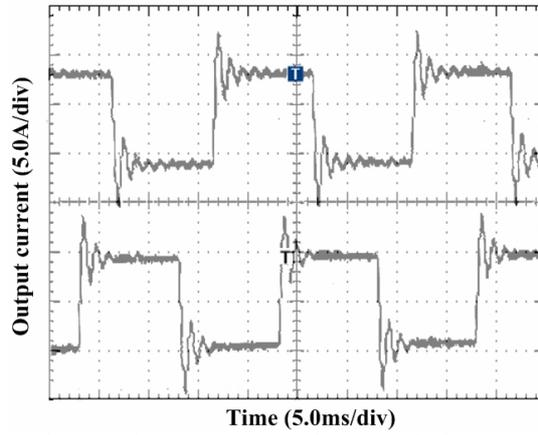
Control method	\hat{e}_{MSE}
Single-neuron quasi-PID control method	0.11%
Quasi-PID control method	2.22%
PI control method	5.75%
Repetitive control method	4.52%
Deadbeat control method	1.93%
Sliding mode control method	3.06%

CONCLUSIONS

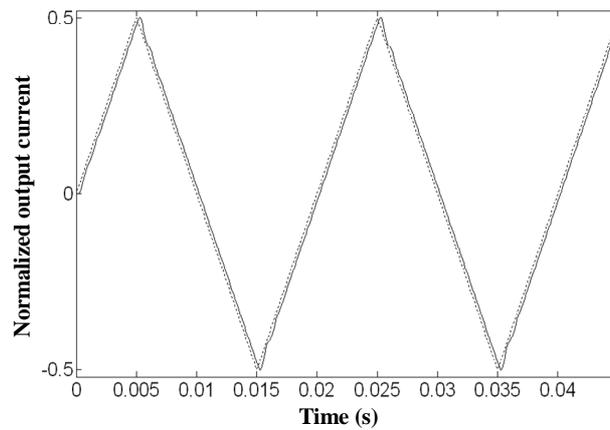
(i) Quasi-PID control method that is directly derived from the circuit topology of SPAs can effectively inhibit the inherent tracking error, and its derivation



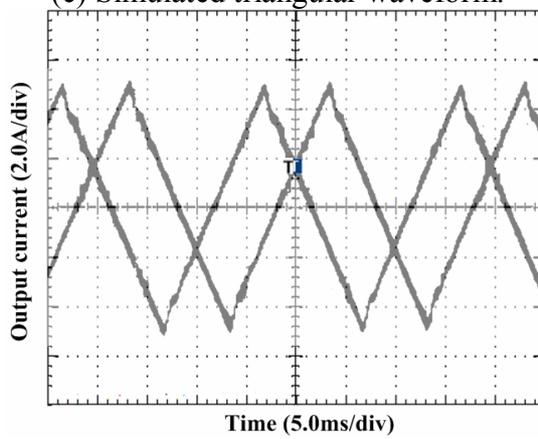
(a) Simulated square waveform.



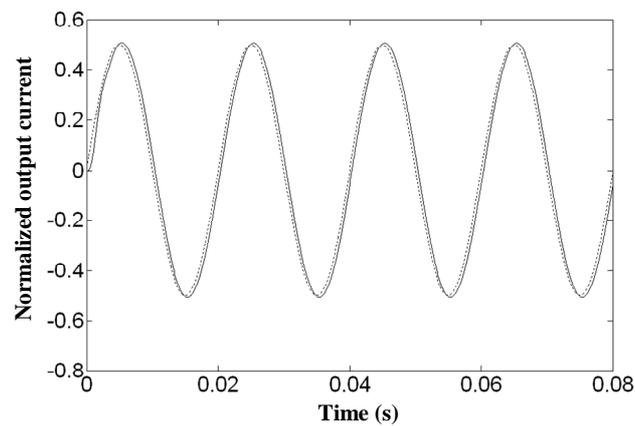
(b) Experimental square waveforms of phases A and B.



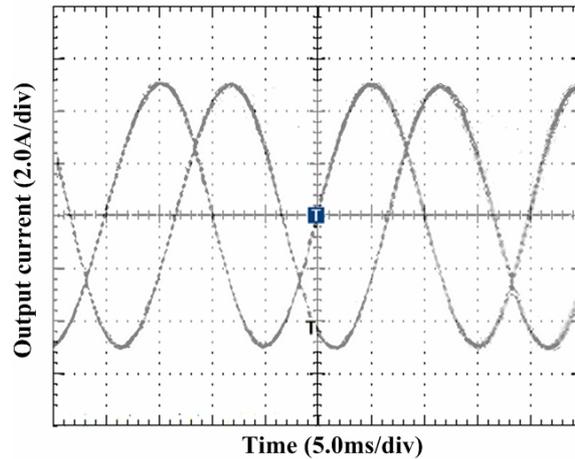
(c) Simulated triangular waveform.



(d) Experimental triangular waveforms of phases A and B.



(e) Simulated sinusoidal waveform.



(f) Experimental sinusoidal waveforms of phases A and B.

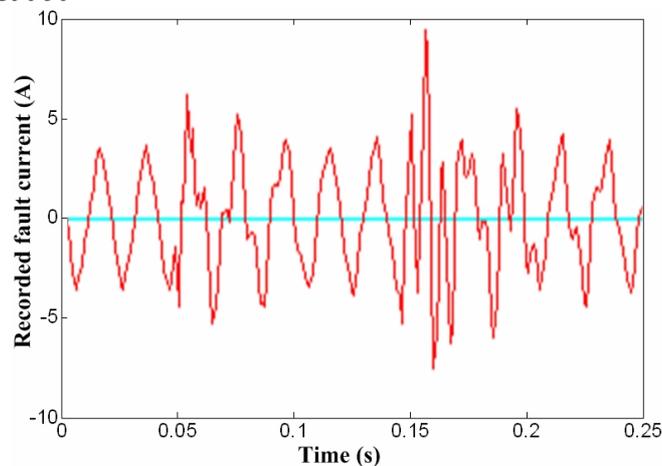
Figure 10. Adaptability of single-neuron quasi-PID control method to different periodic waveforms. Dotted: command signal; solid: output waveform.

reveals an important fact that the quasi-D term is not a real D term, so the strict PID control method is not suitable for SPA control; (ii) quasi-PID control method may suffer from parameter variations and random disturbances, so it needs to be combined with a single neuron to maintain its excellence, and the formation is single-neuron quasi-PID control method; (iii) simulation and experimental results illustrate that single-neuron quasi-PID control method can effectively adapt to different loads, time-variant system parameters, random disturbances and fault waveforms containing various frequency component and nonperiodic components, and maintain a high accuracy; (iv) compared with many other modern control methods, the programming realization of single-neuron quasi-PID control method is very simple, and its computational complexity is very small.

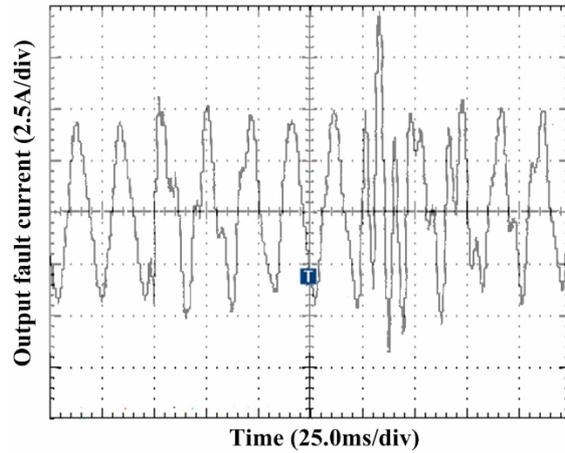
APPENDIX A. CIRCUIT COMPONENTS LIST

Digital Signal Processor (DSP): TMS320 LF2407A

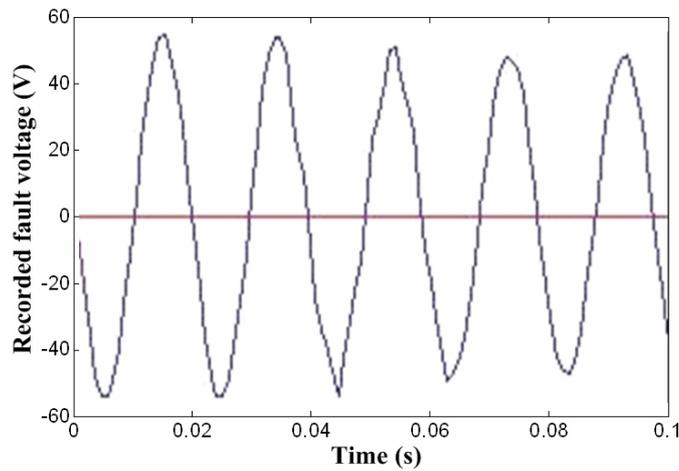
IGBT Module: PM30CSJ060



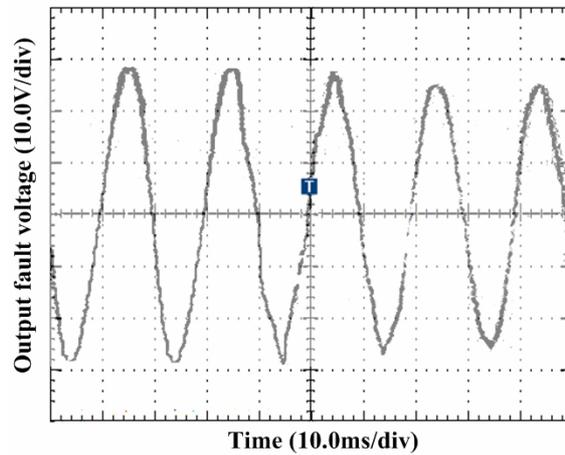
(a) Fault current waveform recorded by DFR.



(b) Experimental fault current waveform of (a).



(c) Fault voltage waveform recorded by DFR.



(d) Experimental fault voltage waveform of (c).

Figure 11. Adaptability of single-neuron quasi-PID control method to fault waveforms recorded by DFRs.

Fast-recovery Free-wheeling Diode: HFA04TB60

DC Energy-storage Capacitor: 4 700 μF

\bar{V}_{dc} : 67 V

L : 1.8 mH

C : 37.6 μF

R : 3 ~ 10 Ω

$T_s: 1 \times 10^{-4} \text{s}$

$f_s: 10 \text{ kHz}$

APPENDIX B. FIGURE OF PROTOTYPE MACHINE



REFERENCES

- [1] Sun XM, Long HL, Liu DC. The design of an electrical fault waveform regenerator. *IEEE International Conference on Electrical Machines and Systems* 2008. **5**:136–141.
- [2] Vasić M, Garcia O, Oliver JA, Alou P, Diaz D, Cobos JA, Gimeno A. ; Pardo JM, Benavente C, Ortega FJ. Efficient and linear power amplifier based on envelope elimination and restoration. *IEEE Transactions on Power Electronics* 2012; **27**(1): 5–9.
- [3] Gong GH, Ertl H, Kolar JW. Novel tracking power supply for linear power amplifiers. *IEEE Transactions on Industrial Electronics* 2008; **55**(2): 684–698.
- [4] El-Habrouk M, Darwish MK, Mehta P. Analysis and design of a new switching power amplifier. *European Power Electronics and Drives Journal* 2000; **10**(3–4): 16–23.
- [5] Kao CH, Lin WP, Chuang YJ. Switching power amplifier with feedback for improving total harmonic distortion. *36th International Conference on Computers and Industrial Engineering* 2006; **1**: 4971–4980.
- [6] Hornik T, Zhong QC. A Current-control strategy for voltage-source inverters in microgrids based on H^∞ and repetitive control. *IEEE Transactions on Power Electronics* 2011. **26**(3):943–952.
- [7] Zhang B, Wang DW, Zhou KL, Wang YG. Linear phase lead compensation repetitive control of a CVCF PWM inverter. *IEEE Transactions on Industrial Electronics* 2008. **55**(4): 1595–1602.
- [8] Hassan TK. A repetitive-PI current controller for boost single phase PFC converters. *Energy and Power Engineering* 2011. **3**:69–78.
- [9] Nishida K, Rukonuzzman M, Nakaoka M. Advanced current control implementation with robust deadbeat algorithm for shunt single-phase voltage-source type active power filter. *IEEE Proceedings of Electric Power Applications* 2004. **151**(3):283–288.
- [10] Mohamed YARI, El-Saadany EF. An improved deadbeat current control scheme with a novel adaptive self-tuning load model for a three-phase PWM voltage-source inverter. *IEEE Transactions on Industrial Electronics* 2007. **54**(2):747–759.
- [11] Zhang Y, Kurihara N. A study of integral sliding mode control with input constraint for engine idling-speed control. *IEEJ Transactions on Electrical and Electronic Engineering* 2012. **7**(2):214–219.
- [12] Benamor A, Chrifi-aloui L, Messaoud H, Chaa-bane M. Sliding mode control, with integrator, for a class MIMO nonlinear systems. *Energy and Power Engineering* 2011. **3**:435–444.
- [13] Guo LP, John YH, Nelms RM. Comparative evaluation of sliding mode fuzzy controller and PID controller for a boost converter. *Electric Power Systems Research* 2011. **81**:99–106.
- [14] Sun XM, Long HL, Liu DC, Li P. Multi-objective optimization design for snubber in switching power amplifier. *International Conference on Electrical Machines and Systems* 2009. **1**:1–6.
- [15] Itoh Y, Hori N, Kamei H. Digital redesign of a nonlinear state-feedback control system based on the principle of equivalent areas. *Proceedings of the SICE Annual Conference* 2004. **1**:350–354.

- [16] Sakamoto T, Hori N. New PWM schemes based on the principle of equivalent areas. *Proceedings of the IEEE International Symposium on Industrial Electronics* 2002. **2**:505–510.
- [17] Franklin GF, Powell JD, Naeini AE. Feedback control of dynamic systems (Fifth Edition). Posts and Telecom Press: Beijing; 2007.
- [18] Kress R. Numerical analysis. Springer-Verlag: New York; 1998.
- [19] Amari SI. Mathematical foundations of Neurocom-puting. *Proceedings of the IEEE* 1990. **78(9)**:1443–1463.
- [20] Haykin S. Adaptive filter theory (Fourth Edition). Upper Saddle River: New Jersey Prentice Hall; 2002.
- [21] Yeung KS. On the Schur-Cohn-Jury Criterion. *International Journal of Control* 1982. **35(3)**: 573–574.
- [22] Jonckheere E, Ma C. A further simplification to Jury's stability test. *IEEE Transactions Circuits and Systems* 1989. **36(3)**:463–464.