International Journal of Robotics and Control Systems
Vol. 5, No. 5, 2025, pp. 2486-2507
ISSN 2775-2658
http://pubs2.ascee.org/index.php/ijrcs

Voltage Regulation Innovation for Photovoltaic DC Bus Using RLS-PZC-Based Autotuning PID with Output Scaler and Discharge Path

Sidik Nurcahyo [a,b,1], Hadi Suyono [a,2,*], Rini Nur Hasanah [a,3], Muhammad Aziz Muslim [a,4]

[a] Department of Electrical Engineering, Faculty of Engineering, Universitas Brawijaya, Malang 65145, Indonesia
[b] Department of Electrical Engineering, Politeknik Negeri Malang, Malang 65141, Indonesia
[1] sidiknurcahyo@student.ub.ac.id; [2] hadis@ub.ac.id; [3] rini.hasanah@ub.ac.id; [4] muh_aziz@ub.ac.id
* Corresponding Author

ARTICLE INFO

Article history
Received August 04, 2025
Revised October 02, 2025
Accepted November 19, 2025

Keywords
Adaptive Control;
Photovoltaic DC Bus;
Renewable Energy Integration;
DC-DC Converter;
Autotuning PID

ABSTRACT

Output voltage of photovoltaic DC bus tends to fluctuate with environment and load dynamics. To prevent undervoltage or overvoltage that may disrupt load operation, a voltage regulator is required. This article proposes a new voltage regulator using a buck converter and RLS-PZC-based autotuning PID with output scaler and discharge path. The research contribution includes minimizing voltage fluctuation and maintaining the transient response characteristic. Periodically, RLS (Recursive Least Squares) identifies the buck converter's transfer function by processing its input-output signals, and then PZC (Pole Zero Cancellation) uses the RLS result to update PID parameters in producing an overshoot-free DC bus voltage that settles in a specified duration. To prevent the output from fluctuating due to input voltage variation, the input voltage is read by a sensor, and the output scaler uses the reading to adjust the PID output in keeping forward path gain at a particular value. A discharge path is activated during negative PID output to further reduce spikes caused by load shedding. The proposed method is simpler and deterministic, unlike metaheuristic approaches. It is implemented in a Matlab script and C program code running on an 8-bit 16-MHz microcontroller in a Proteus simulation. Results show that RLS-PZC can identify load changes and retune PID parameters in 40.2 ms. The output fluctuation due to input voltage variation can be significantly reduced by the output scaler, where its ITAE (Integral of Time-weighted Absolute Error) is only 9.27% of regular PID. Employing a discharge path can improve voltage regulation by reducing the regular PID spikes up to 73.6% when a 5-ohm load is shed from a 50-ohm default load.

1. Introduction

Photovoltaic (PV) systems have become a prospective research topic in the last decades [1]-[3]. PV systems are one of the promising alternative solutions to the provision of environmentally friendly electrical energy [4], [5]. The main problem of the PV systems is that their maximum power point changes according to environmental conditions, such as solar irradiance, temperature, and partial

shading phenomenon [6]-[8]. This problem has been successfully solved using a solar tracker [9]-[12] and an MPPT (Maximum Power Point Tracker). MPPT is a more attractive solution than solar trackers due to lower investment and higher power efficiency. There are three types of MPPT, including classical, intelligent, and optimization [13]. Some examples are P&O [14]-[19], INC [20], [21], Neuro-Fuzzy [22]-[28], MCA-FOCV [29], RP-FOSMC [30], PSO [31]-[33], GTO [34], GWO [35]-[37], AFO [38], SM [39], Snake [40], and Adaptive Jaya [41]. All these MPPT algorithms work by adjusting the duty cycle of a power converter connecting the PV system to the load so that PV power can be delivered to the load optimally. Such a working principle causes the MPPT output voltage to fluctuate following the environmental conditions and load demand [42]-[44]. If the voltage fluctuations are not handled properly, they can lead to overvoltage and undervoltage, which in turn disrupt the load operation [45]. To address this problem, a voltage regulator is needed [46]-[48].

In the case of a PV system, the voltage regulator is a device that receives an input voltage from the MPPT and produces an output voltage on the DC bus with an amplitude maintained at a certain level [49]. As the MPPT output voltage is higher than the desired DC bus voltage, the voltage regulator can be developed using a buck converter and its controller [50]. The regulator should combat two disturbances, including the non-constant input voltage as a result of the MPPT and the dynamic load as a result of demand. Based on the mathematical analysis described later in this paper, changes in the input voltage will cause changes in the DC gain of the buck converter, while changes in the load will cause changes in the damping constant of the buck converter. Thus, the buck converter parameters vary with input voltage and load. To obtain good and consistent time response performance, an adaptive or autotuning controller is required. This type of controller is equipped with capabilities to tune its parameters online so that the control signal will always be correct, and hence the resulting response will be well maintained in any input voltage and load conditions [51].

If the controller is PID, the general and well-known controller, then the autotuning mechanism can be developed using metaheuristic techniques like AOA [52], Fuzzy [53], [54], data-driven [55], Neural Network [56], ADIWASO PSO [57], RL [58], Deep RL [59], and Relay [60]. These techniques are indeterministic, rather complicated, and require more power consumption while executing the algorithm. This article proposes an autotuning RLS-PZC (Recursive Least Square – Pole Zero Cancellation) PID (Proportional Integral Derivative) with an output scaler and discharge path. This choice is motivated by the success of the RLS-PZC PI on first-order plants [61]. The RLS-PZC PID is a deterministic and simpler controller, thus less power consumption, unlike adaptive PID-based voltage regulators tuned using CNN [62], SSA-PSO [63], BAT [64], CI [65], or ELM [66]. The proposed controller is not only able to update the PID parameters quickly using the PZC (Pole Zero Cancellation) based on the buck converter's transfer function estimated using the RLS (Recursive Least Squares), but also inherently can identify load changes and hence will maintain its time response characteristics. The output scaler will address the input voltage variation effect on output voltage fluctuation by maintaining the forward path gain using a dynamic gain directly calculated based on input voltage measurement. The discharge path is for reducing overshoot caused by load shedding. For simplicity, the proposed system does not integrate a storage battery as described in [67].

The main contribution of this research lies in the RLS-PZC-based autotuning PID equipped with an output scaler and discharge path to regulate the DC bus voltage of the PV system at a certain value with significantly reduced fluctuation (undervoltage/overvoltage) and to maintain its time response characteristics even though both input voltage and load change dynamically.

2. Method

The proposed diagram for photovoltaic DC bus voltage regulation is shown in Fig. 1. The buck converter (G) receives an uncertain load (R) as a result of power demand and indeterministic input voltage (V_i) as a result of MPPT working. PID controller (C) is assigned to maintain the DC bus voltage (V_o) to always be equal to the setpoint (V_s) by processing the error signal (E) into the control signal (M) using PID parameters as a result of PZC, including proportional gain (K_p), integral gain

(K_i), and derivative gain (K_d). Determination of PID parameters by PZC is based on the RLS-estimated buck converter parameters (K, ζ, ω_n), the maximal PID output (M_{max}), and the required time constant (τ).

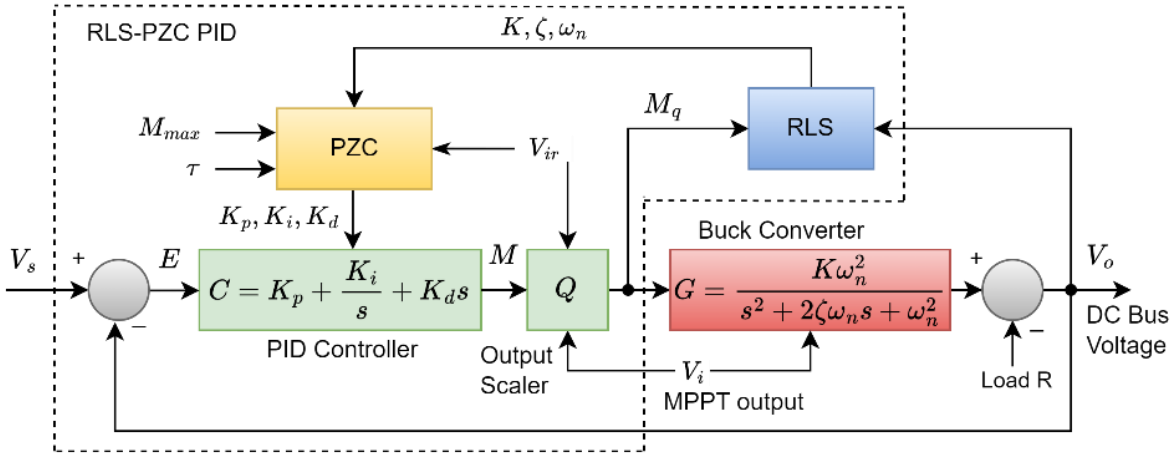


Fig. 1. Control diagram for buck converter voltage regulator using RLS-PZC PID with output scaler

Research steps to verify the proposed method include: (a) obtaining the buck converter transfer function and verifying that input voltage affects DC gain and that load affects the buck converter damping constant; (b) determining the PZC formula and proving that PZC-tuned PID can produce overshoot-free response and that its settling time is as designed; (c) deriving the RLS algorithm and proving that RLS can estimate buck converter parameters; (d) confirming that the output scaler is able to suppress the influence of changes in input voltage to output voltage fluctuations; and (e) integrating the RLS, PZC, PID, output scaler, and discharge path to construct a complete adaptive controller and evaluating its capacity to maintain the DC bus output voltage and suppress overshoot significantly. Verifications are done by Matlab simulation, and several are by Proteus simulation, as Proteus can simulate a microcontroller running C program code for some algorithms as described in [68]-[70].

2.1. Transfer Function Derivation for Buck Converter

A buck converter's transfer function is required to prove that its DC gain depends on the input voltage, and its damping constant depends on the load. The transfer function is also needed for deriving the PZC formula. Consider a buck converter circuit, as shown in Fig. 2.

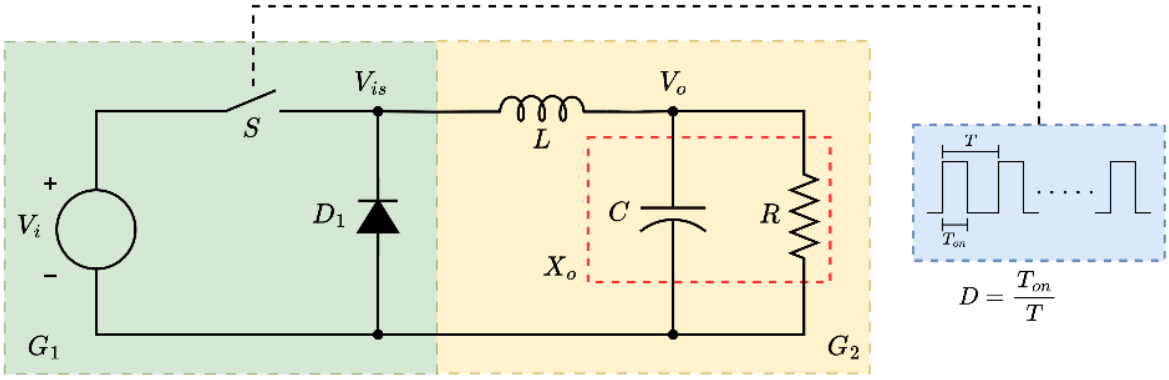


Fig. 2. Buck converter circuit

If S is a negligible impedance and high-frequency semiconductor switch, then the average output voltage of the switch (V_{is}) can be expressed as

$$V_{is} = DV_i \tag{1}$$

where D is a duty cycle defined as

$$D = T_{on}/T \tag{2}$$

T_{on} is the duration of the switch S for closing its contact, and T is the switch period, where *T=1/f*. Meanwhile, the relation between output voltage V_o and V_{is} can be expressed as

$$V_o/V_{is} = X_o/(sL + X_o) \tag{3}$$

The output impedance X_o can be calculated as follows.

$$X_o = R/(sCR + 1) \tag{4}$$

Substituting (4) into (3) yields

$$V_o/V_{is} = [1/(CL)]/[s^2 + s/(\mathrm{CR}) + 1/(\mathrm{CL})] \tag{5}$$

Eliminating V_{is} in (5) using (1) yields buck converter transfer function.

$$V_o/\mathrm{D} = [V_i/(\mathrm{CL})]/[s^2 + s/(CR) + 1/(\mathrm{CL})] \tag{6}$$

If (6) is compared with the following second-order transfer function.

$$\mathrm{G} = \mathrm{Y}/\mathrm{U} = [K\omega_n^2]/[s^2 + 2\zeta\omega_n s + \omega_n^2] \tag{7}$$

Then DC gain (K), damping constant (ζ), and the natural frequency (ω_n) are as follows.

$$K = V_i;\ \omega_n = 1/\sqrt{CL};\ \zeta = 1/\left(2R\sqrt{C/L}\right) \tag{8}$$

This proves that DC gain depends on input voltage (V_i) and damping constant depends on load (R).

2.2. PZC Derivation for Tuning the PID Parameters

Prior to applying the proposed autotuning PID on the DC bus regulation, the PZC formula needs to be derived first. Equation (7) can be rewritten.

$$G = b_0/(s^2 + a_1 s + a_0) \tag{9}$$

where

$$b_o = K\omega_n^2;\ a_o = \omega_n^2;\ a_1 = 2\zeta\omega_n \tag{10}$$

Using the following PID transfer function.

$$C = (K_d s^2 + K_p s + K_i)/s \tag{11}$$

The closed-loop transfer function is as follows:

$$G_c = \left(K_d s^2 + K_p s + K_i\right)b_0/\left[s(s^2 + a_1 s + a_0) + \left(K_d s^2 + K_p s + K_i\right)b_0\right] \tag{12}$$

The denominator of G_c can be expanded.

$$PK1 = s^3 + s^2(a_1 + K_d b_0) + s\left(a_0 + K_p b_0\right) + K_i b_0 \tag{13}$$

To result in an overshoot-free response, two poles of *PK1* need to be canceled using zero of G_c. This implies that *PK1* need to be converted into the following *PK2A,* and its expanded form is *PK2B*.

$$\begin{gathered} PK2\mathrm{A} = \left(K_d s^2 + K_p s + K_i\right)(\mathrm{s}/\mathrm{K_d} + \alpha); \\ PK2\mathrm{B} = s^3 + s^2\left(K_d\alpha + K_p/K_d\right) + s\left(K_p\alpha + K_i/K_d\right) + K_i\alpha \end{gathered} \tag{14}$$

Equating the coefficient s^0 of *PK1* and *PK2B* yields:

$$\alpha = b_0 \tag{15}$$

Equating the coefficient s^1 of *PK1* and *PK2B* and replacing α with b_0 yields:

$$K_i = a_0 K_d \tag{16}$$

Equating the coefficient s^2 of *PK1* and *PK2B* and replacing α with b_0 yields:

$$K_p = a_1 K_d \tag{17}$$

Applying *PK2A* as the denominator of G_c results in:

$$G_c = 1/[\mathrm{s}/(K_d b_0) + 1] \tag{18}$$

Hence, the closed-loop time constant (τ) can be expressed.

$$\tau = 1/(K_d b_0) \tag{19}$$

If the buck converter's settling time ($t_s = 4\tau$) is selected at design, then:

$$K_d = 1/(\tau b_0) \tag{20}$$

Finally, the PID parameters (Kp, Ki, and Kd) can be calculated using (17), (16), (20), and buck converter parameters.

2.3. RLS Algorithm for Estimating the Buck Converter Parameters

The RLS algorithm is required to estimate buck converter parameters through buck converter input-output signals. To implement the RLS algorithm on 8-bit low-speed microcontrollers, complex variable (*s*) of (9) is need to be replaced with the following backwards euler [71].

$$s \leftarrow (1 - z^{-1})/T \tag{21}$$

where *T* is sampling time, thus resulting in the following difference equation:

$$y = (b_o T^2 u + (2 + a_1 T) y z^{-1} - y^{-2})/(a_1 T + a_o T^2 + 1) \tag{22}$$

where *u* and *y* are the input and output of the buck converter, respectively. The parameters to be estimated include b_o, a_o, and b_1. For simplicity, defined:

$$y = pu + qy_1 + ry_2 \tag{23}$$

where $p = b_o T^2/N;\ q = (2 + a_1 T)/N;\ r = 1/N;\ N = a_1 T + a_o T^2 + 1$. Variables y_1 and y_2 represent yz^{-1} and yz^{-2}, respectively. The parameters of *p, q,* and *r* can be estimated through RLS [72], where its algorithm is as shown in Table 1.

If those parameters have been found, then plant parameters a_1, a_o, and b_o can be calculated using $a_1 = (q/r - 2)/T;\ a_o = (1/r - 1 - aT)/T^2; b_o = p/(rT^2)$. This algorithm includes 11 steps, but only 9 steps are required for each iteration, as the first is an initialization done once, and the last step is a repeat instruction for the RLS algorithm. Step 9 is optional; it is required to compare the estimated output to the actual output. Step 10 is for calculating PID parameters, and it is mandatory, as the final goal of RLS is to get PZC-tuned PID parameters that can maintain performance.

2.4. PID with Output Scaler

The simplified form of Fig. 1 without RLS and PZC is shown in Fig. 3. Based on (6), DC gain for *G* is changed when V_i changes; hence, the forward path gain of this control system is also changed with V_i. Meanwhile, the PWM gain that receives the carrier frequency f_c is fixed for various V_i. If the default PID parameters were calculated using PZC with input voltage V_{ir} and the actual input voltage

V_i differs from V_{ir}, then to ensure that the default PID parameters still comply with any value of V_i, the forward path gain needs to be fixed using the output scaler Q having a dynamic gain.

$$Q = D/M = V_{ir}/V_i \tag{24}$$

where V_i is measured and V_{ir} is determined at design. This solution is different from standard feedforward compensation [73], where it adjusts forward gain directly based on measured buck converter input voltage, rather than based on the estimated plant parameters. It also differs from gain normalization, where input voltage is used for scaling the adaptation gain, not for fixing the forward path gain. PV's uncertainties will only contribute to variation in MPPT output voltage that in turn causes variation in the buck converter's input voltage or plant DC gain and finally changes forward path gain. As forward path gain is fixed by the output scaler and load change is detected recursively by RLS, then in principle, global stability will be ensured.

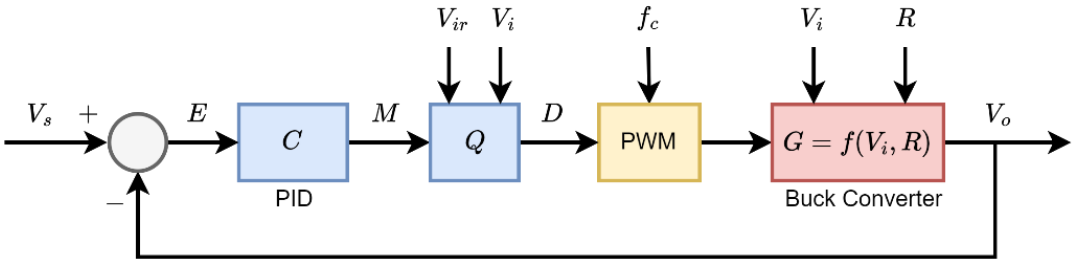


Fig. 3. Closed-loop PID control with output scaler Q

Table 1. RLS algorithm for buck converter

Step	Low level Operations	Remaks
1	$y_1 = 0; y_2 = 0; p_{11} = p_{22} = p_{33} = \delta = 1000$; others zero; $p = 0; q = 0; r = 0;\ \lambda = 0.98$	Initialise variables $P = \begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p33 \end{bmatrix} \hat{\theta} = \begin{bmatrix} p \\ q \\ r \end{bmatrix};\ \Phi = \begin{bmatrix} u \\ y_1 \\ y_2 \end{bmatrix}$
2	$[u, y]$ = readInputOutput()	Read [u,y], update Φ with *u*.
3	$e = y - (pu + qy_1 + ry_2)$	Calculate error $e = y - \hat{y}$
4	$\beta = \lambda + u(p_{11}u + p_{12}y_1 + p_{13}y_2) + y_1(p_{21}u + p_{22}y_1 + p_{23}y_2) + y_2(p_{31}u + p_{32}y_1 + p_{33})$ $k_1 = (p_{11}u + p_{12}y_1 + p_{13}y_2)/\beta$ $k_2 = (p_{21}u + p_{22}y_1 + p_{23}y_2)/\beta$ $k_3 = (p_{31}u + p_{32}y_1 + p_{33}y_2)/\beta$	Calculate K $K = \frac{P\Phi}{\lambda + \Phi^T P \Phi}$
5	$p = p + ek_1;\ q = q + ek_2; r = r + ek_3$	Calculate dummy parameters (*p, q, r*) $\hat{\theta} = \hat{\theta} + Ke$
6	$p_{11} = (p_{11} - (k_1 u p_{11} + k_1 y_1 p_{21} + k_1 y_1 p_{31}))/\lambda$ $p_{12} = (p_{12} - (k_1 u p_{12} + k_1 y_1 p_{22} + k_1 y_1 p_{32}))/\lambda$ $p_{13} = (p_{13} - (k_1 u p_{13} + k_1 y_1 p_{23} + k_1 y_1 p_{33}))/\lambda$	Update P $P = (P - K\Phi^T P)/\lambda$
7	$y_2 = y1;\ y_1 = y$	Update y_1 and y_2
8	$a_1 = (q/r - 2)T;\ a_o = (1/r - 1 - aT)/T^2; b_o = p/(rT^2)$	Calculate actual parameter (*a, b, c*)
9	Optional: $\hat{y} = pu + qy_1 + ry_2$	Calculate estimated output $\hat{y}$, for comparison with the actual output y; $\hat{y} = \Phi(\mathrm{k})\hat{\theta}$
10	$K_d = \frac{1}{\tau V_i c}; K_i = bK_d; K_p = aK_d$	Calculate PID parameters using PZC
11	repeat to step 2	End of one iteration

3. Results and Discussion

3.1. Verification for the Buck Converter Transfer Function

To verify the obtained buck converter's transfer function in (6), a Matlab simulation is prepared as shown in Fig. 4. This simulation consists of a transfer function of (6) and a buck converter circuit

(Fig. 2) with parameters of R=100Ω, L=330mH, C=47μF, PWM (Pulse Width Modulation) frequency fp=25kHz, duty cycle D=0.5, and MOSFET (Ron=17$m\Omega$, Rd=0.01Ω, snubber Rs=100kΩ, Cs=none). These parameters are populated from a Matlab script or typed manually from the Matlab command line to ensure the same values are applied to the circuit and transfer function. Responses for both systems are sent to a scope for comparison. Running this simulation results in a response as shown in Fig. 5. This response proves that (6) is correct, as the transfer function output (Vom) fits the buck converter output (Vo).

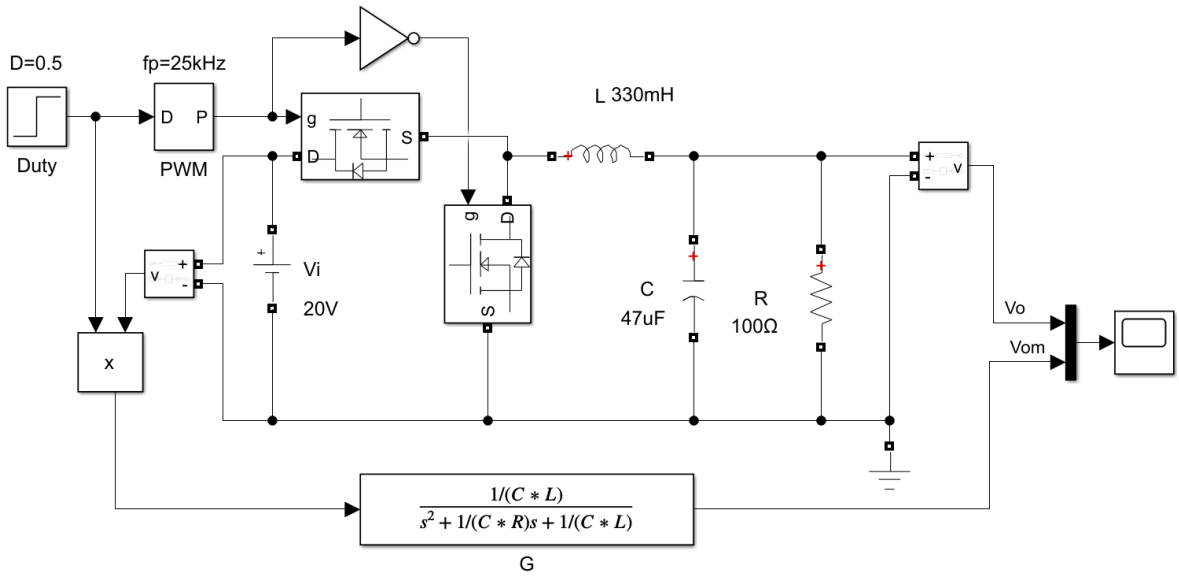


Fig. 4. Matlab simulation for verification the transfer function

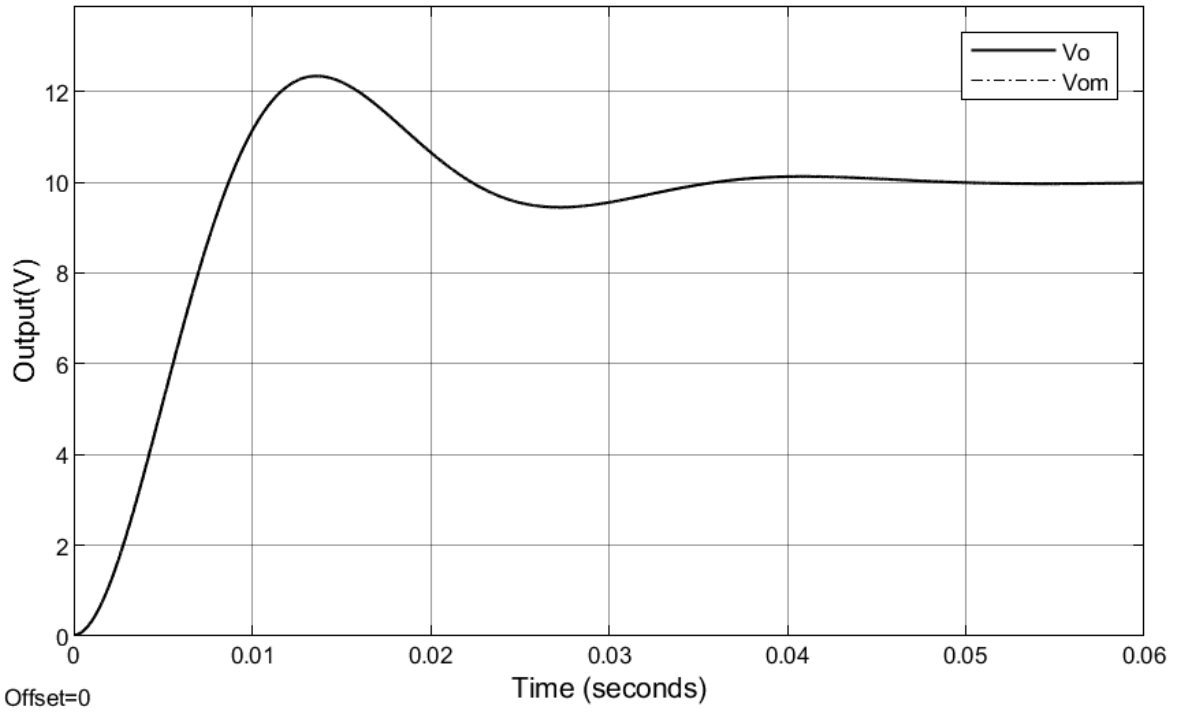


Fig. 5. Output comparison of transfer function and buck converter when $R = 100\Omega$, $D = 0.5$, and $Vi = 20V$

To know the load effect on the buck converter dynamics, a response comparison is made for three loads, R = [5, 15, 100] ohms, and with two input voltages, Vi = [30, 60] V, as shown in Fig. 6.

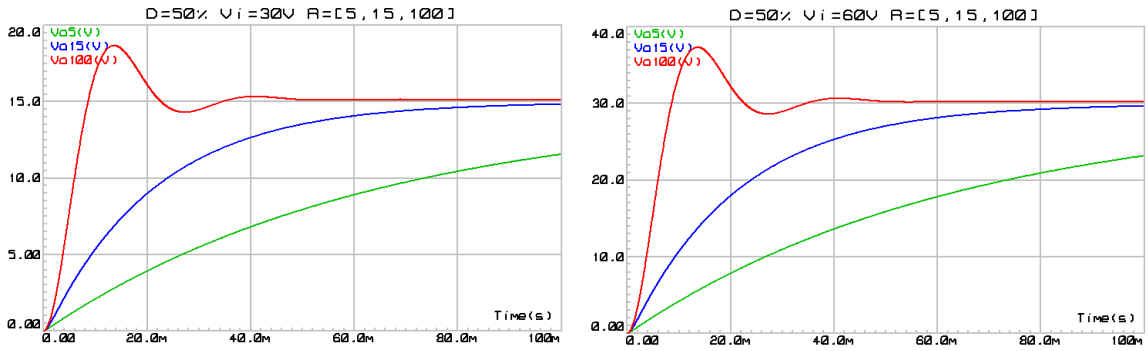
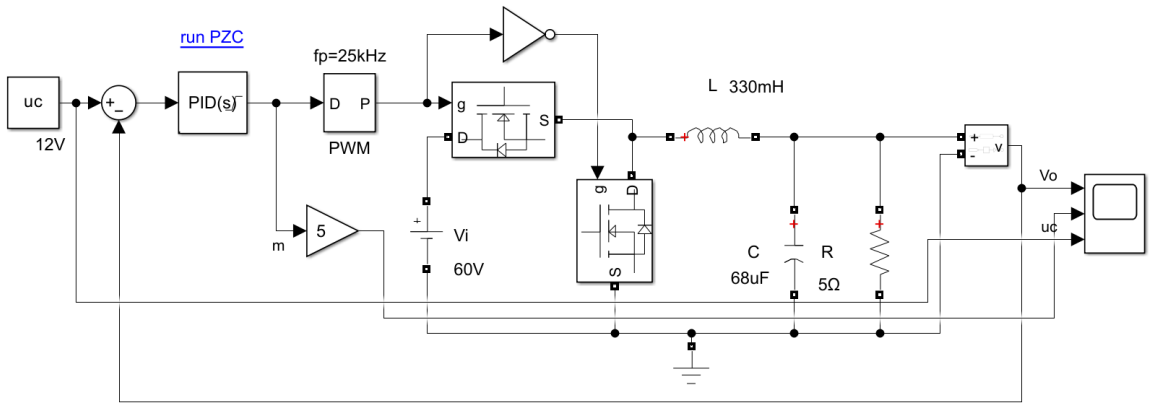


Fig. 6. Buck converter response for various R when Vi=30V (left) and when Vi=60V (right)

These plots show that different load R results in different shapes of response. This complies with (8), where R contributes to the damping constant (ζ). The left response settles at 15V as its input voltage is 30V, and the right response settles at 30V as its input voltage is 60V. This fact indicates that the input voltage (V_i) determines the steady-state value or its DC gain. These results prove that the buck converter transfer function and its response shape are influenced by load (R) and input voltage (V_i). If it is required that the regulation performance be maintained at any load and input voltage, it is mandatory to employ an autotuning controller on the buck converter.

3.2. Verification for PZC

To verify the PZC formula, a PZC script shown in Fig. 7 (b) and a PZC-tuned PID closed-loop control system for a buck converter shown in Fig. 7 (a) are constructed using MATLAB.



(a) Test circuit for verifying the PZC formula

```
% PZC_script.mlx
Vi=60;                  % input voltage
R=5; L=330e-3;          % load and inductor
C=68e-6;                % capacitor
uc = 12;                % setpoint
K = Vi;                 % plant DC gain
ts = 60e-3              % required settling time
tau = ts/4
ao = (1.0/(C*L))
a1 = (1.0/(C*R))
bo = (K*ao)
Kd = (1.0/(tau*bo))
Ki = (ao*Kd)
Kp = (a1*Kd)
```

(b) PZC script for calculating the PID parameters

Fig. 7. Matlab simulation for PZC-tuned PID

Executing the PZC script will populate Kp = 0.073, Ki = 1.111, and Kd = 2.493e-5 in MATLAB's workspace so they are available for the PID block. Simulating the diagram of Fig. 7 (a) results in a response as shown in Fig. 8.

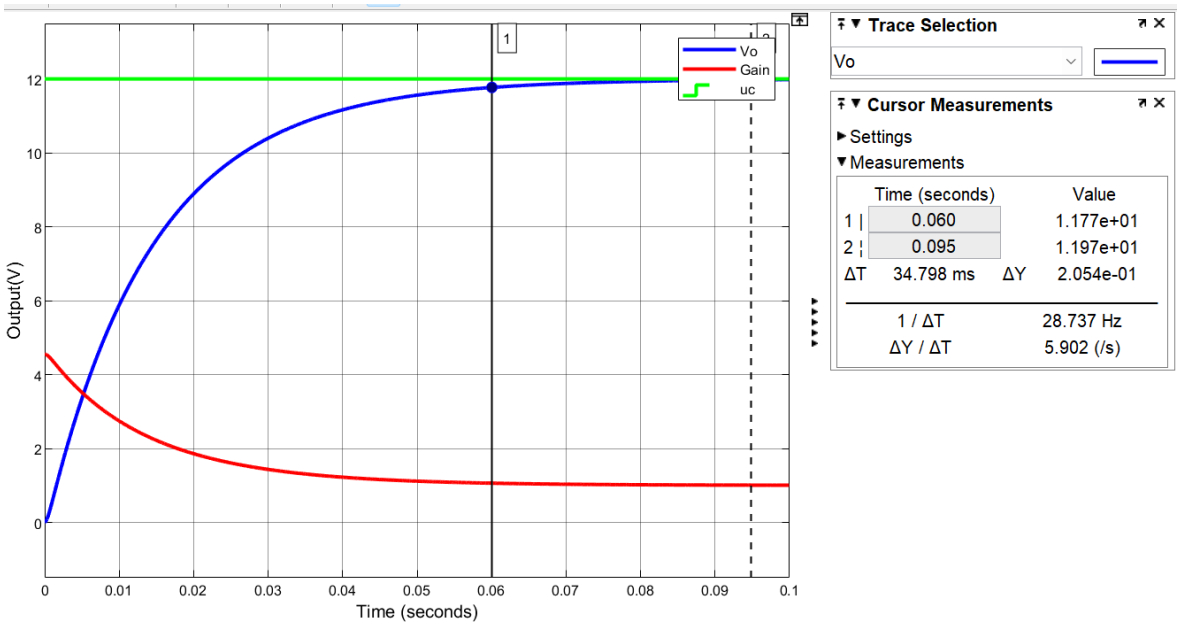


Fig. 8. Response for PZC-tuned PID in Matlab

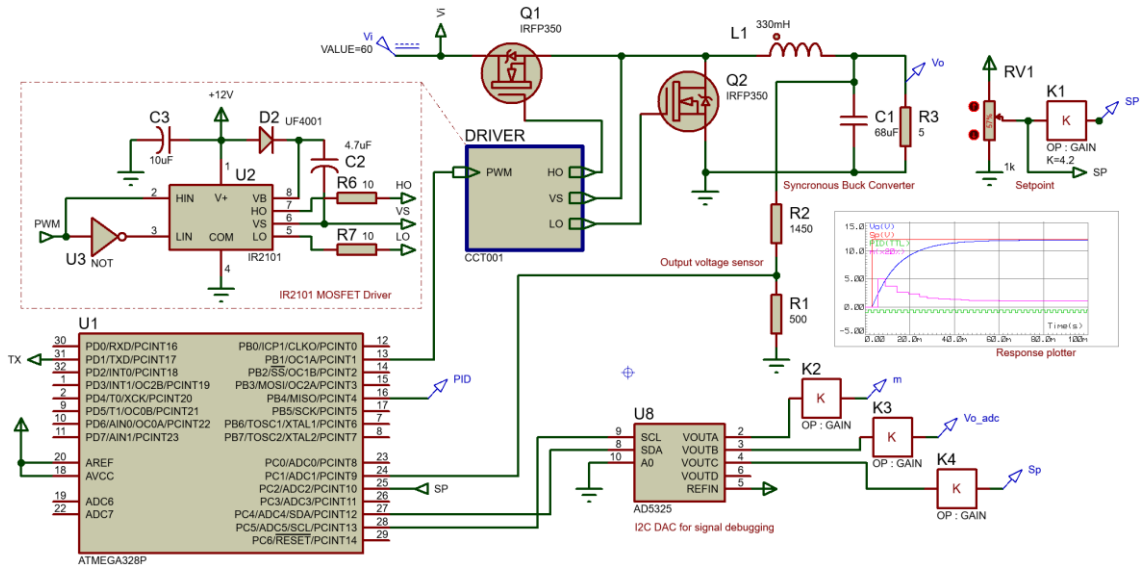
This result shows that the PZC successfully tunes the PID parameters as the resulting response settles (enters 11.77 V, or 98% of 12 V) in 60 ms, as defined by the PZC script line 7, and it does not contain overshoot. This confirms that the PZC formula is correct. To know the applicability of PZC in more real situations, a Proteus simulation is prepared as shown in Fig. 9 (a). ATmega328 (U1) is an 8-bit microcontroller assigned to run C program code for PZC-tuned digital PID. The important part of this C-code is shown in Fig. 9 (b). Line 47 defines the required settling time, ts=60 ms. Lines 48-52 define input voltage and buck converter parameters. Lines 56-63 calculate the PID parameters using the PZC formula. Lines 89-90 define space-time (1 ms) and PID execution time (1.69 ms), respectively. The space-time is provided to execute instruction lines 82-84 for sending several debugged signals, while the PID execution time is for executing the *pid* function (lines 92-120). A timer overflow interrupt service routine, lines 122-126, is used to ensure time accuracy.

This setting enables the PID algorithm to be executed periodically with a sampling time of 2.69 ms. U1 reads setpoint SP through ADC2 (lines 105-106), reads feedback through ADC1 (lines 103-104), and sends PID output in the form of a PWM signal through OC1A (line 117). The I2C DAC AD5325 (U8) is assigned to populate debugged signals, including PID output (m), setpoint (Sp), and buck converter output (Vo_adc), respectively, using instruction lines 82-84. A response plotter is provided to display those debugged signals. The PB4 pin is set at the beginning of *pid* function (line 99) and reset at the end of *pid* function (line 119) to enable measuring the PID execution duration through the response plotter. Running this simulation results in a response shown in Fig. 10. This plot consists of four signals, including buck converter output (Vo), setpoint (Sp), execution indicator (PID), and duty cycle output (m). As can be seen, the Vo curve is equal to the Vo curve in Fig. 8, where it settles in 60 ms and does not contain overshoot. This result indicates that the PZC-tuned PID is successfully implemented on a microcontroller and its computational fidelity is guaranteed.

3.3. Verification for the RLS-PZC

To verify the effectiveness of the RLS algorithm in estimating the buck converter parameters and calculating the PID parameters using PZC, a Proteus simulation is prepared as shown in Fig. 11. The RLS-PZC algorithm of Table 1 is implemented in a C program code run on U2 Atmega328 (16 MHz clock) with a sampling time of 20.1 ms, managed by a timer overflow interrupt. To explore whether

the input signal type determines the estimation result, several input signals are provided. They can be selected from the output of K5, K6, K7, or K8 by uniquely renaming the required output terminal to *u*. U2 reads the input signal (*u*) through ADC1. The resulting ADC data is converted to a PWM (Pulse Width Modulation) signal and sent to the buck converter through OC1A. R1 and R2 are used to scale down the output signal (Vo) to become *yc*; hence, its maximum amplitude is 5V. This signal is read from ADC2. Meanwhile, the load R in Fig. 7 is implemented by R3 of 10 ohms. The load is changed dynamically from 10 ohms to 5 ohms at t=1s and back to 10 ohms at t=1.5s through activating and deactivating a relay, RL1, using a step signal called *R_en*. The resulting PZC-tuned PID parameters are sent to an I2C DAC AD5325 (U1) to enable displaying them in a graphical plot. Running the simulation several times with different input signals yields responses shown in Fig. 12.



(a) Testing circuit for PZC-tuned PID in Proteus

```c
#define ts   60e-3f  //required settling time
#define Vi   60.0    //nominal voltage to calcu
//plant parameters
#define R    5.0f    //converter load R
#define C    68e-6f  //converter filter C
#define L    330e-3f //converter inductor L

#define Gi   (Vi/Vi)
//calculate PID gains using PZC
#define K    (Gi*Vi)
#define tau  (ts/4.0)     //4.4 for settling in
#define ao   (1.0/(C*L))
#define a1   (1.0/(C*R))
#define bo   (K*ao)
#define Kd   (1.0/(tau*bo))
#define Ki   (ao*Kd)
#define Kp   (a1*Kd)
volatile float m;         //output PID
volatile float adc_y;
volatile float adc_uc;

#define ADC_CH_Y    1
#define ADC_CH_UC   2

int main(){
    i2c_init();
    sei();
    PORTC = _BV(PC4) | _BV(PC5);
    adc_init();
    pwm_init();
    uart_init(9600);
    timer0ovf_init();
    adc_uc = adc_read(ADC_CH_UC);
    DDRB |= (1<<PB4);
    while (1){
        dac(0, 0x0FFF*m/M);
        dac(2, adc_uc*4);
        dac(1, adc_y*4);
    }
    return 0;
}
```

```c
#define PID_SPACE_MS 1
#define PID_ELAPSED_TIME_MS 1.69f //got by evaluating PB4 in graph pl

void pid(){
    const uint8_t N_PID = PID_SPACE_MS;
    static uint8_t i = N_PID;
    if(++i<N_PID)
        return;
    i = 0;
    TCCR0B = 0;
    PORTB |= _BV(PB4);
    const float T = (PID_SPACE_MS + PID_ELAPSED_TIME_MS)/1000.0;
    static float e_1 = 0.0;
    static float I = 0.0;
    adc_y = adc_read(ADC_CH_Y);
    float y = (float)adc_y*(5.0/sensor_gain);
    adc_uc = adc_read(ADC_CH_UC);
    float uc = 21.0*adc_uc;
    float e = (uc - y)/1023.0;

    I += T*e;
    m = (Kp*e + Ki*I + Kd*(e-e_1)/T)*M;
    e_1 = e;

    if(I>=M) I=M;
    if(m>=M) m = M;
    else if(m<0.0) m=0;

    pwm_writeA(m);
    TCCR0B = (TCNT0_CS(TCNT0_PRESCALE)<<CS00);  //run timer
    PORTB &= ~(1<<PB4);
}

ISR(TIMER0_OVF_vect)
{
    TCNT0 = 255+1-TCNT0_PRELOAD; //62 step to overflow--->Tick = 0.99
    pid();
}
```

(b) Snipped C-code for the PZC-tuned PID

Fig. 9. Proteus simulation for PZC-tuned PID

As depicted in the left responses, the estimated output (Voe) tends to follow the output (Vo). This indicates that RLS works as expected. When a new load R19 is applied at t=1s, the effective load R

will be 5 ohms and cause the PID parameters to change slightly. When the load is recovered to 10 ohms at 1.5s by disconnecting R19, the PID parameters are also recovered to the initial value before t=1s. It is also shown that all parameter curves among these trials are similar with minor differences, including settling time and spike height. Their characteristics are summarized in Table 2.

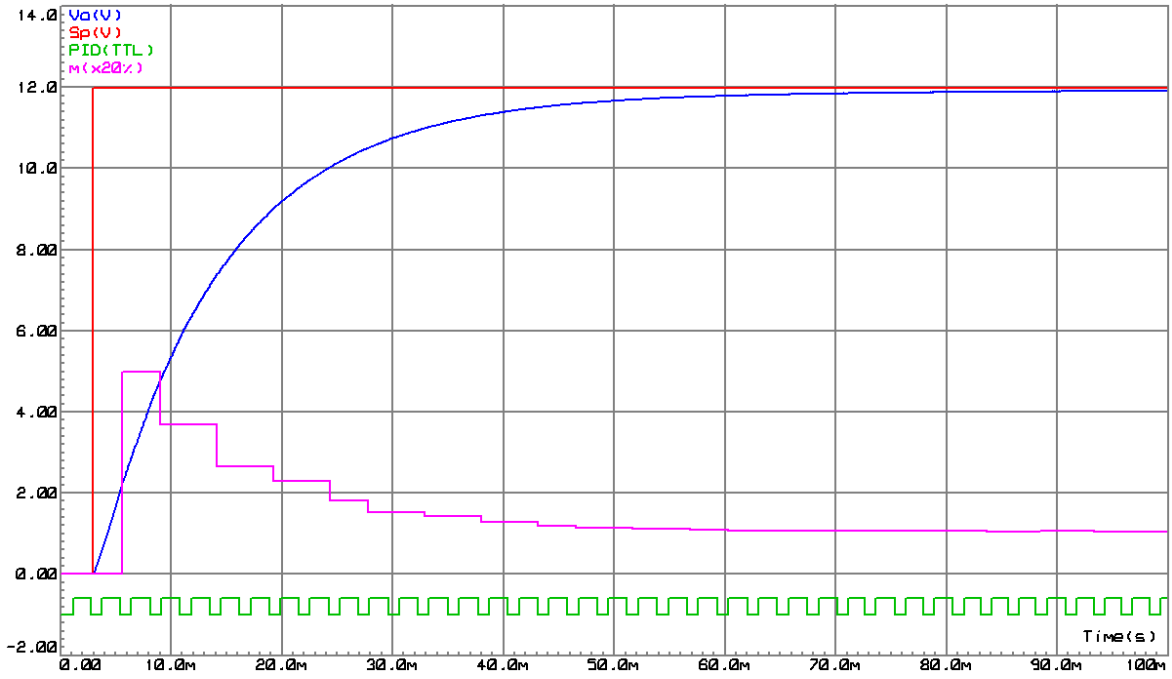


Fig. 10. Response for PZC-tuned PID in Proteus

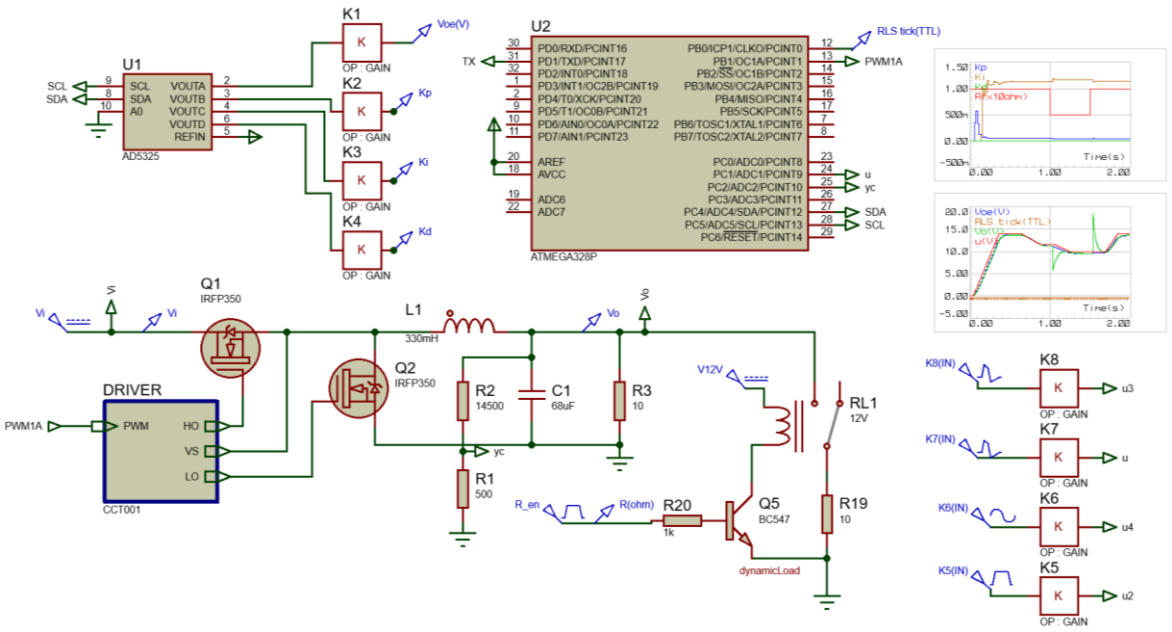
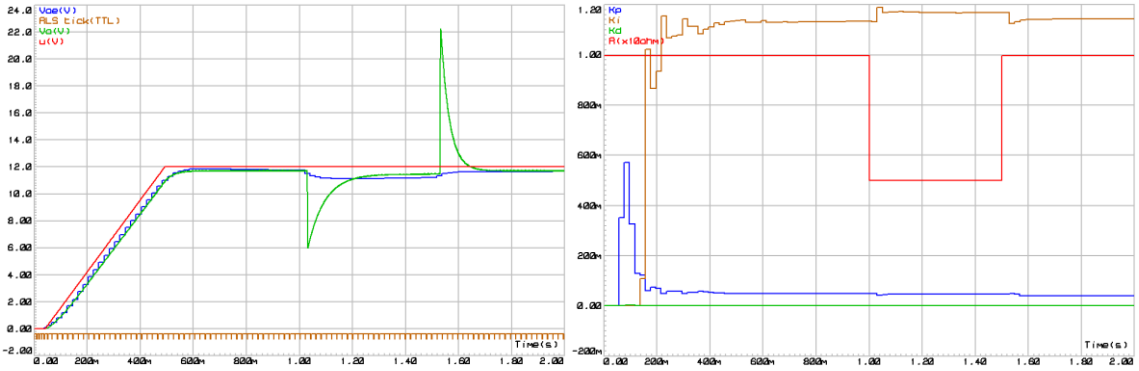


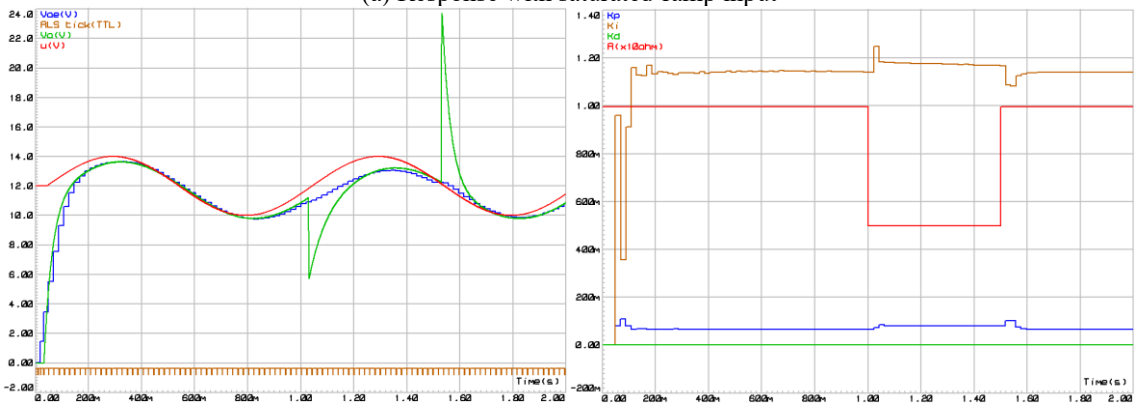
Fig. 11. RLS circuit test with selectable inputs and load change R

These data show that the best input signal is a combinational ramp, as the next settling time is shorter, the PID parameter spike is lower, and the PID parameters are closer to the ideal value (depicted in parentheses under the column headings of Kp, Ki, and Kd). Unfortunately, this input signal results in a longer initial settling time. However, all these input signals are suitable to use, as they can result in the PID parameters that are settled at particular values near to the ideal values. These results signify that the RLS-PZC algorithm has been successfully implemented to estimate buck

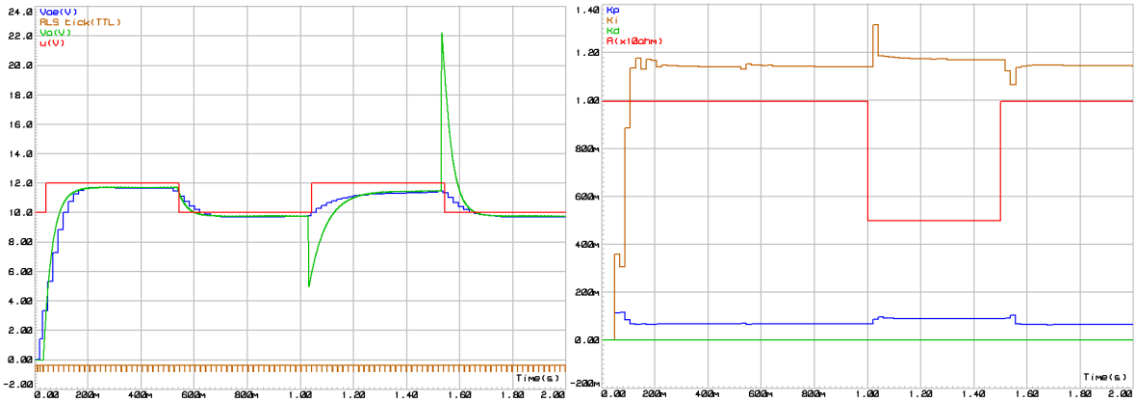
converter parameters (including load changes) and to calculate PID parameters in two sampling times (40.2 ms).



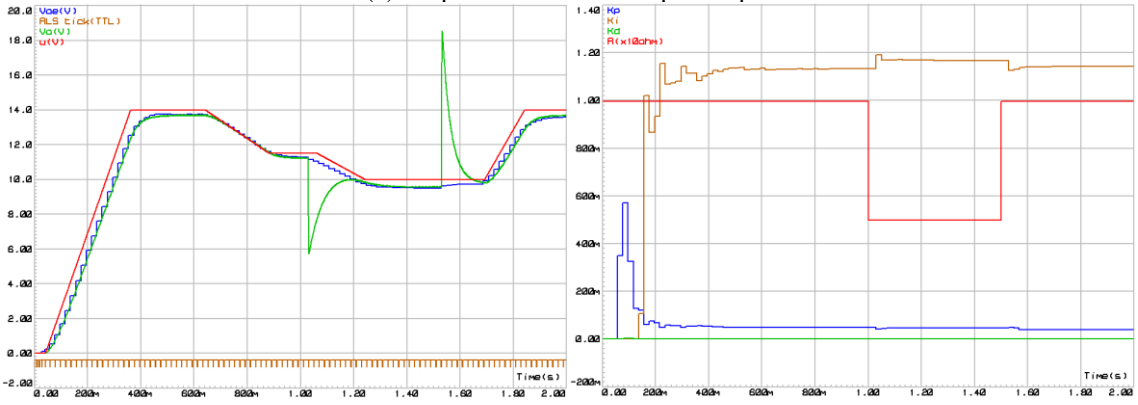
(a) Response with saturated-ramp input



(b) Response with shifted-sinusoidal input



(c) Response with shifted-square input



(d) Response with a combinational saturated-ramp input

Fig. 12. RLS-PZC response with different inputs and load change

Table 2. Response characteristic for RLS-PZC-based PID parameter tuning

Input Type, Response Figure	Settling time (s)		PID parameters's spike	PZC-Tuned PID parameters at R=10 ohm / R=5 ohm		
	initial	next		Kp (0.037/0.073)	Ki (1.11/1.11)	Kd (2.49e-5/2.49e-5)
Saturated ramp, Fig. 12 (a)	0.6	0.04	low	0.051/0.043	1.13/1.19	138e-6/138e-6
Shifted sinus, Fig. 12(b)	0.2	0.2	moderate	0.067/0.081	1.14/1.17	138e-6/138e-6
Shifted square, Fig. 12 (c)	0.2	0.3	high	0.069/0.089	1.14/1.17	138e-6/138e-6
Combinational ramp, Fig. 12(d)	0.5	0.04	low	0.049/0.046	1.13/1.17	138e-6/138e-6

3.4. Verification for the PID Output Scaler

The output scaler contribution is verified using a Proteus simulation, as shown in Fig. 13. This simulation consists of two closed-loop digital PID control systems for buck converters. PID parameters for both control systems are tuned using the PZC formula, and their PID algorithms are implemented in C program code, as shown in Fig. 9 (b).

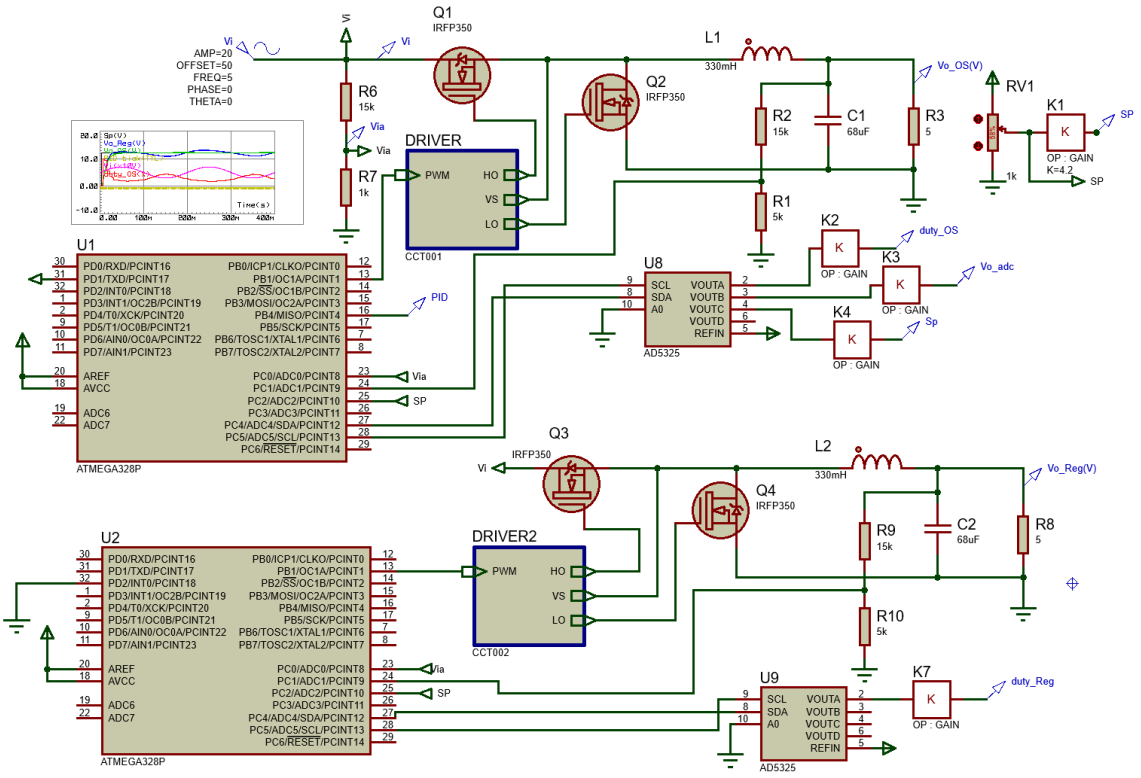


Fig. 13. Proteus simulation for testing the PID output scaler

The first microcontroller (U1) is assigned to run the digital PID with output scaler (PID-OS), where its PID output is scaled by Q before being sent to the DRIVER. The second microcontroller (U2) serves as a regular digital PID (PID-R) without the output scaler. Both PIDs read their setpoint from ADC2, read their feedback from ADC1, and deliver their manipulated signal (in the form of a PWM signal) through OC1A. The main difference between the two is that PID-OS reads the input voltage of the buck converter (Vi) from ADC0, but PID-R does not. This signal is used to maintain the forward path gain as defined in (24). Both PID algorithms are executed regularly, leveraging a timer overflow interrupt to implement a constant sampling time of 3.1 ms. In this setup, the buck converter receives dynamic input voltage (Vi) consisting of a 50 V DC source plus a sinusoidal signal (amplitude 20 V, frequency 5 Hz, and phase 0°). Running this simulation results in a response as shown in Fig. 14.

There is a significant difference between the PID-OS response (Vo_OS) and the PID-R response (Vo_Reg), where Vo_Reg oscillates around its setpoint (Sp), but Vo_OS does not. Calculating their

ITAE (Integral of Time-weighted Absolute Error) during steady state (t=60-400 ms) results in 0.2188 for PID-R and 0.0203 for PID-OS, or only 9.27% of the PID-R. This result indicates that the output scaler can significantly reduce output fluctuation affected by input voltage dynamics.

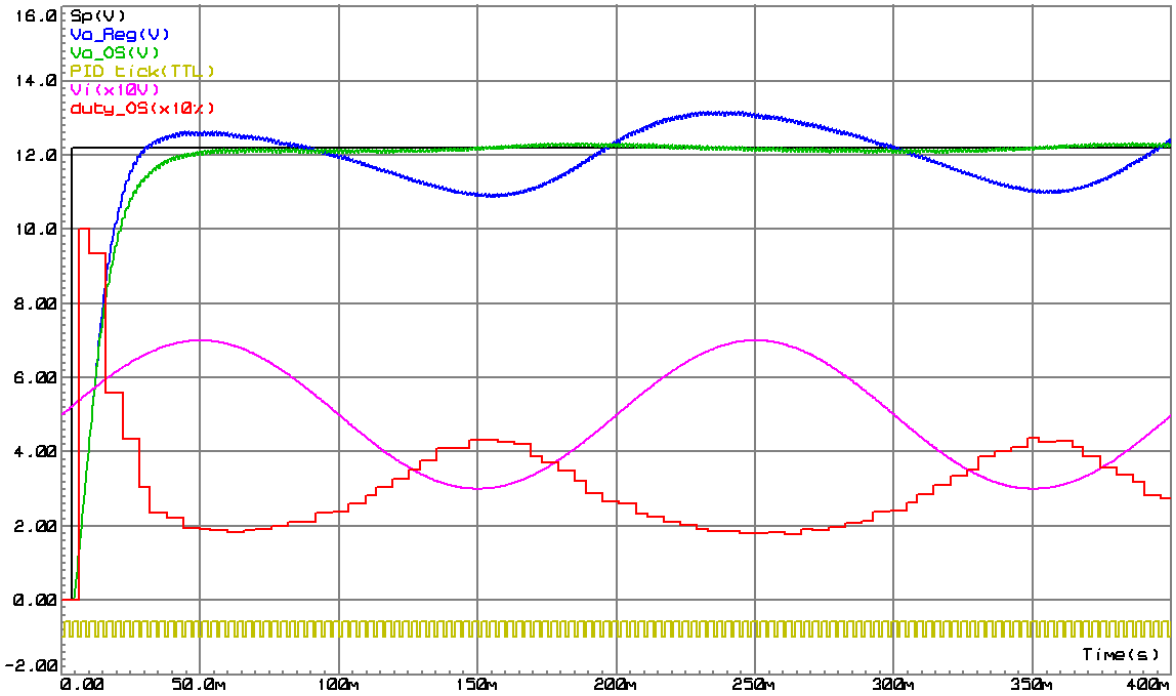


Fig. 14. Response comparison between regular PID and output scaler-based PID

3.5. Performance Evaluation for Proposed System

To evaluate the proposed method, RLS-PZC, PID, output scaler, and discharge path are integrated, and its response is compared to regular PID using Matlab simulation as shown in Fig. 15.

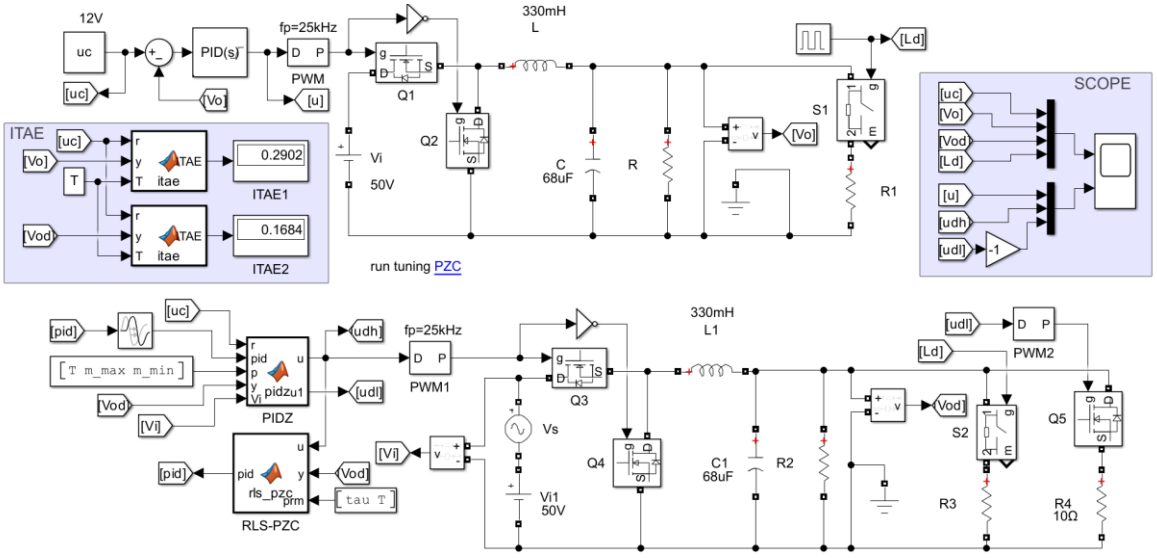


Fig. 15. Matlab simulation for performance evaluation of the proposed method

The upper diagram is for regular PID, where its parameters are initially tuned using the PZC formula. Meanwhile, the bottom diagram is for the proposed system, where PIDZ parameters are online auto-tuned using RLS-PZC. Simulation conditions, including sampling period T, noise level, varying input voltage, and load changes, are described inherently by the circuit and accompanying scripts. The Matlab script for simulation initialization and PIDZ is shown in Fig. 16.

```
% Simulation initialization
Vi=50;      % input voltage
R=5; R1=20;% load (default & new)
L=330e-3;  % buck converter L
C=68e-6;   % buck converter C
uc = 12;   % setpoint
m_max = 1; % max actuator output
m_min = -1;% min actuator output
K = Vi;    % dc gain plant
ts = 60e-3 % settling time
tau = ts/4 % time constant

ao= (1.0/(C*L))
a1= (1.0/(C*R))
bo= (K*ao)
Kd= (1.0/(tau*bo))
Ki= (ao*Kd)
Kp= (a1*Kd)
T = 1e-4    % sampling time
```

```
function [u,u1] = pidz(r,pid, p,y,Vi)

persistent I;
persistent e1;
if isempty(I)
    I = 0;
    e1 = 0;
end
[Kp, Ki, Kd] = struct('x', num2cell(pid)).x;
[T, m_max, m_min] = struct('x', num2cell(p)).x;
e = r - y;
I = I + T*e;
u = Kp*e + Ki*I  + Kd*(e-e1)/T;
u = u*50.0/Vi;

if u>m_max u=m_max; end
if u<m_min u=m_min; end
if I>m_max I=m_max; end
if I<m_min I=m_min; end

u1=0;
if u<0
    u1 = abs(u);
    u=0;
end
e1 = e;
```

Fig. 16. Matlab script for simulation initialization (left) and PIDZ (right)

In the left script, load and sampling time are defined in lines 3 and 18, respectively. Referring to the right script, lines 11-13 calculate PID output *u,* and line 14 scales the signal *u* using the output scaler gain of (24). For reducing overshoot due to load shedding, a discharge path (including R4, Q5, and PWM2 of Fig. 15) is activated when the PIDZ output is negative, as defined by lines 22-25. The PIDZ receives its parameters through *pid* variable sent by RLS-PZC with a script, as shown in Fig. 17.

```
function pid   = rls_pzc(u,y,prm)

persistent y1; persistent y2; persistent P;
persistent theta_est; persistent err; persistent Kp;
persistent Ki; persistent Kd; persistent i; delta = 0.1
lambda = 0.9999; N = 4;
if isempty(y1)
    i=0; Kp = 0.1; Ki = 1.5; Kd = 0;
    err = ones(1,N); y1 = 0; y2 = 0;
    P = delta * eye(3);
    theta_est = zeros(3,1);
end

phi = [u; y1; y2];
e = y - phi' * theta_est;
K_k = (P * phi) / (lambda + phi' * P * phi);
theta_est = theta_est + K_k * e;
P = (P - K_k * phi' * P) / lambda;

y2 = y1;
y1 = y;
```

```
p = theta_est(1);
q = theta_est(2);
r = theta_est(3);

[tau, T] = struct('x', num2cell(prm)).x
a1 = -(q/r+2)/T;
ao = -(1+r*(a1*T+1))/(r*T^2);
bo = -(p/(r*T^2));

i = i + 1;
if i>N i=1; end
err(1,i) = e*e/N;
s=0;
for j=1:N s = s + err(1,j); end

if (s)<1e-3
    Kd= (1.0/(tau*bo));
    Ki= 1.5*(ao*Kd);
    Kp= 3*(a1*Kd);
end

pid=[Kp Ki Kd];
```

Fig. 17. Matlab script for RLS algorithm

Lines 39-41 are for updating PID parameters only when estimation is in steady state or when the sum of squared error for the last four samples is small enough, e.g., 1e-3. This scenario is for reducing excessive fluctuation caused by Kp and Ki spikes, as shown by the right graphs of Fig. 12 at t=1s and t=1.5s. To improve regular PZC, Kp and Ki are increased by trial and error, 3 and 1.5 times, respectively (see lines 40-41). Performance evaluation is done for 0.6s with the PIDZ and RLS sampling time of 0.1ms. During evaluation, input voltage (Vi) for the upper diagram is kept constant at 50V, while input voltage for the lower diagram is varied in a sinusoidal pattern, Vi = 50 + 10 sin(2πt). Loads for both diagrams are equal, and they are changed at t=0.2s and restored to their default value at t=0.4s by activating/deactivating switches S1 and S2 to connect/disconnect R1 and R3,

respectively. To get a fair comparison, the default load R is set to equal to R2, and the new load R1 is set to equal to R3. Besides comparing voltage peaks caused by load shedding, evaluation is also done by comparing ITAE values calculated using the ITAE block containing a script as shown in Fig. 18.

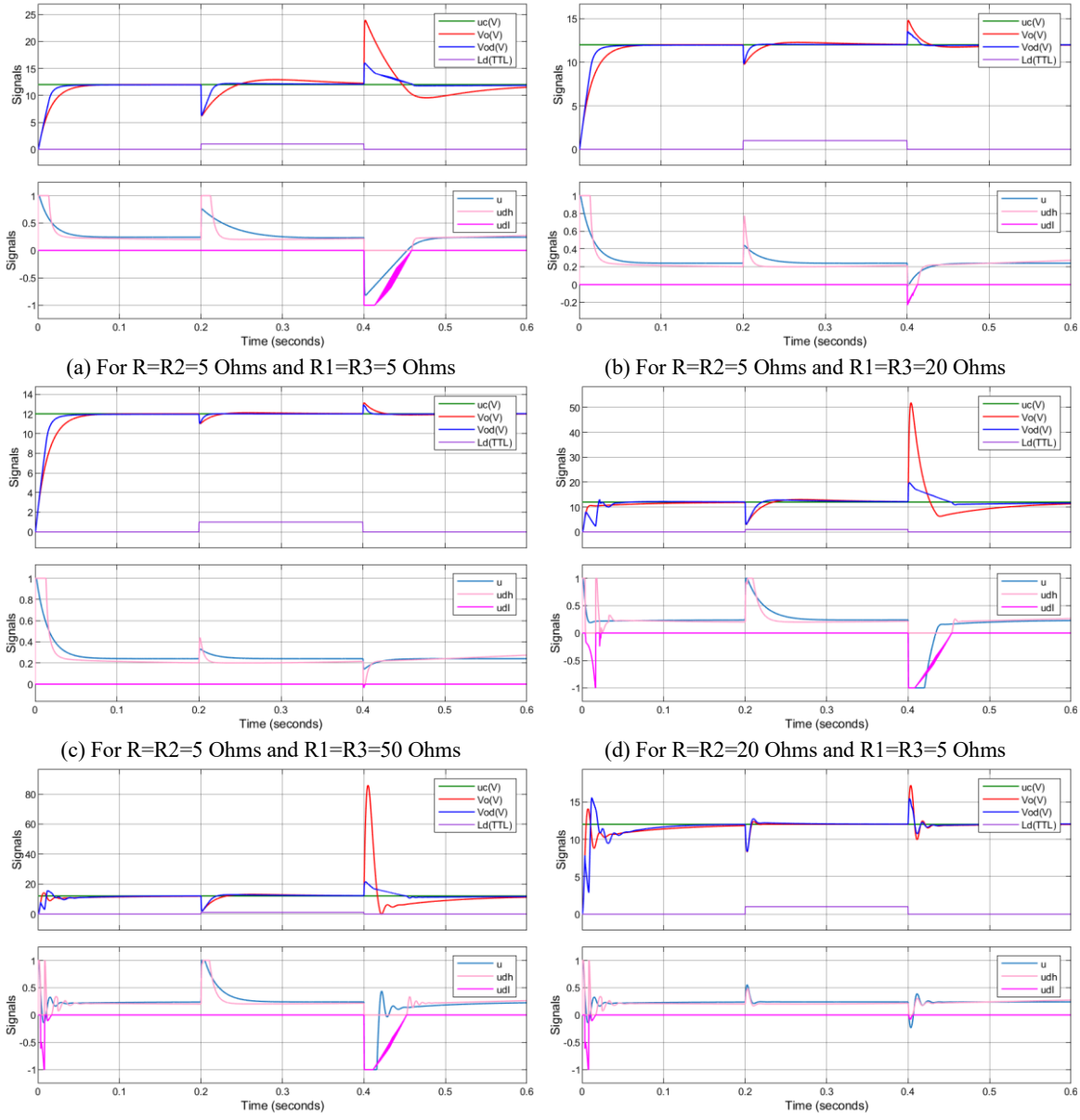
```
function ITAE  = itae(r,y,T)

persistent itae;
if isempty(itae)
    itae = 0;
end

itae = itae + abs(r-y)*T;
ITAE= itae;
```

Fig. 18. Matlab script for calculating ITAE

Running the simulation several times with different combinations of default loads (R, R2) and new loads (R1, R3) results in responses as shown in Fig. 19.



(a) For R=R2=5 Ohms and R1=R3=5 Ohms

(b) For R=R2=5 Ohms and R1=R3=20 Ohms

(c) For R=R2=5 Ohms and R1=R3=50 Ohms

(d) For R=R2=20 Ohms and R1=R3=5 Ohms

(e) For R=R2=50 Ohms and R1=R3=5 Ohms

(f) For R=R2=50 Ohms and R1=R3=50 Ohms

Fig. 19. Response comparison between proposed (Vod) and regular PID (Vo)

When a new load (R1 and R3) is applied to the buck converter at t=0.2s, both output voltages drop, but the proposed method can recover the output faster, and while recovering, it does not lead to overshoot except when the new load R3 is 50 ohms, as shown in Fig. 19 (f). When a new load (R1 and R3) is shed at t=0.4s, output voltage increases, but the proposed method can recover the output faster and does not lead to undershoot except when the new load R3 is 50 ohms, as shown in Fig. 19 (f). Regarding fluctuation due to load shedding, Fig. 19 (a), Fig. 19 (d), and Fig. 19 (e) at t=0.4s show that the regular PID cannot reduce voltage fluctuation significantly, but the proposed method can. As summarized by Table 3, the proposed method results in lower peaks than the regular PID. The highest peak reduction is 73.6% when a new load of 5 ohms is shed from the default load of 50 ohms. These results signify the superiority of the proposed method against load changes.

Table 3. Responses characteristics

Responses	R, R2 (Ω)	R1, R3 (Ω)	ITAE			Load Shedding Peak		
			itaeVo	itaeVod	Difference	peakVo (V)	peakVod (V)	Reduction (%)
Fig. 19(a)	5	5	0.8561	0.3202	0.5359	23.96	15.89	33.7
Fig. 19(b)	5	20	0.2902	0.1684	0.1218	14.81	13.64	7.9
Fig. 19(c)	5	50	0.2249	0.1441	0.0808	13.12	12.90	1.7
Fig. 19(d)	20	5	1.3110	0.5997	0.7113	51.93	19.97	61.5
Fig. 19(e)	50	5	1.7310	0.6295	1.1015	81.56	21.53	73.6
Fig. 19(f)	50	50	0.2551	0.2535	0.0016	17.17	15.33	10.7

The superiority can also be assessed through ITAE values; a lower ITAE denotes better performance because the buck converter output is less likely to deviate from its setpoint. Table 3 shows that the ITAE value of the proposed method (itaeVod) is always less than the regular PID (itaeVo). These results indicate the promised improvement regarding the voltage regulation.

4. Conclusion

This article proposes an RLS-PZC-based adaptive PID controller equipped with an output scaler and discharge path for improving voltage regulation performance of photovoltaics DC bus. The main objective of this innovation is to reduce the ITAE value by suppressing overshoot and undershoot due to load change and input voltage variation. The simulation results show that RLS can track the buck converter's parameter, PZC can retune PID parameters to produce an overshoot-free response with the desired settling time, the output scaler successfully reduces fluctuation regarding the input voltage changes, and the discharge path can reduce overshoot caused by load shedding. The PZC formula becomes a theoretical contribution for PID tuning. The output scaler and discharge path also become additional contributions to the voltage regulators. To reduce ITAE further, modified Kp and Ki are used, where their new values are 3 times and 1.5 times, respectively. The multipliers selection requires more elaboration for justification. To our knowledge, this is the first work to demonstrate a discharge-path-enhanced RLS-PZC PID with output scaler implemented on an 8-bit microcontroller with Proteus simulation. However, it requires real testing to determine their applicability in real conditions. This result contributes to the adaptive control system, especially on voltage regulation for photovoltaic DC buses.

Suggested future work may include improving the PID autotuning method, robustness test, real hardware experiments, and implementing its hardware details. Enhancements are required to improve its reliability, such as increasing the sampling frequency and upgrading the microcontroller to ESP32 (4MB memory, 270MHz) and STM32 (2MB, 250MHz) to reduce latency. Scalability, or scaling up the buck converter's power rating, is also a challenge, as it relates to the selection of the power switch and sensor. Integrating a storage battery using a bi-directional converter into the photovoltaic DC bus may be an alternative for further suppressing voltage fluctuation.

Author Contribution: All authors contributed equally to this paper and have approved the final paper.

Funding: This research did not receive external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

[1] T. Kirchartz, G. Yan, Y. Yuan, B. K. Patel, D. Cahen, and P. K. Nayak, "The state of the art in photovoltaic materials and device research," *Nature Reviews Materials*, vol. 10, no. 5, pp. 335-354, 2025, https://doi.org/10.1038/s41578-025-00784-4.

[2] A. O. Ali *et al*., "Advancements in photovoltaic technology: A comprehensive review of recent advances and future prospects," *Energy Conversion and Management: X*, vol. 26, p. 100952, 2025, https://doi.org/10.1016/j.ecmx.2025.100952.

[3] A. J. Alrubaie, A. Al-Khaykan, R. Q. Malik, S. H. Talib, M. I. Mousa and A. M. Kadhim, "Review on MPPT Techniques in Solar System," *2022 8th International Engineering Conference on Sustainable Technology and Development (IEC)*, pp. 123-128, 2022, https://doi.org/10.1109/IEC54822.2022.9807500.

[4] T. Rehman *et al*., "Global perspectives on advancing photovoltaic system performance—A state-of-the-art review," *Renewable and Sustainable Energy Reviews*, vol. 207, p. 114889, 2025, https://doi.org/10.1016/j.rser.2024.114889.

[5] J. Gómez-Catasús *et al*., "Solar photovoltaic energy development and biodiversity conservation: Current knowledge and research gaps," *Conservation Letters*, vol. 17, no. 4, p. e13025, 2024, https://doi.org/10.1111/conl.13025.

[6] H. Karimi, A. Siadatan and A. Rezaei-Zare, "A Hybrid P&O-Fuzzy-Based Maximum Power Point Tracking (MPPT) Algorithm for Photovoltaic Systems Under Partial Shading Conditions," *IEEE Access*, vol. 13, pp. 86046-86056, 2025, https://doi.org/10.1109/ACCESS.2025.3533314.

[7] A. Pedroza-Díaz, P. M. Rodrigo, Ó. Dávalos-Orozco, E. De-la-Vega, and Á. Valera-Albacete, "Review of explicit models for photovoltaic cell electrical characterization," *Renewable and Sustainable Energy Reviews*, vol. 207, p. 114979, 2025, https://doi.org/10.1016/j.rser.2024.114979.

[8] R. B. Bollipo, S. Mikkili, and P. K. Bonthagorla, "Hybrid, optimal, intelligent and classical PV MPPT techniques: A review," *CSEE Journal of Power and Energy Systems*, vol. 7, no. 1, pp. 9-33, 2021, https://doi.org/10.17775/CSEEJPES.2019.02720.

[9] E. Akpinar, G. Gül Katircioglu, and M. Das, "Effects of solar tracking on different types of solar panels; experimental study for thermal and photovoltaic types," *International Journal of Hydrogen Energy*, vol. 144, pp. 611-620, 2025, https://doi.org/10.1016/J.IJHYDENE.2025.02.155.

[10] A. M. A. S. Azad *et al*., "Harnessing the sun: Framework for development and performance evaluation of AI-driven solar tracker for optimal energy harvesting," *Energy Conversion and Management: X*, vol. 26, p. 100990, 2025, https://doi.org/10.1016/j.ecmx.2025.100990.

[11] M. Ali, H. Suyono, M. A. Muslim, M. R. Djalal, Y. M. Safarudin, and A. A. Firdaus, "Determination of the parameters of the firefly method for PID parameters in solar panel applications," *SINERGI*, vol. 26, no. 2, p. 265, 2022, https://doi.org/10.22441/sinergi.2022.2.016.

[12] R. Sadeghi, M. Parenti, S. Memme, M. Fossa, and S. Morchio, "A Review and Comparative Analysis of Solar Tracking Systems," *Energies*, vol. 18, no. 10, p. 2553, 2025, https://doi.org/10.3390/en18102553.

[13] R. B. Bollipo, S. Mikkili, and P. K. Bonthagorla, "Critical Review on PV MPPT Techniques: Classical, Intelligent and Optimisation," *IET Renewable Power Generation*, vol. 14, no. 9, pp. 1433-1452, 2020, https://doi.org/10.1049/iet-rpg.2019.1163.

[14] A. B. Djilali, A. Yahdou, H. Benbouhenni, A. Alhejji, D. Zellouma, and E. Bounadja, "Enhanced perturb and observe control for addressing power loss under rapid load changes using a buck–boost converter," *Energy Reports*, vol. 12, pp. 1503-1516, 2024, https://doi.org/10.1016/j.egyr.2024.07.032.

[15] H. S. Chatterjee and S. N. Mahato, “Modified Perturb and Observe-based MPPT control of MHPCS for single-phase power distribution positioned in remote locations,” *Electrical Engineering*, vol. 106, no. 6, pp. 6893-6909, 2024, https://doi.org/10.1007/s00202-024-02390-z.

[16] X. Li, Y. He, and M. Li, “Research on Photovoltaic Maximum Power Point Tracking Control Based on Improved Tuna Swarm Algorithm and Adaptive Perturbation Observation Method,” *Energies*, vol. 17, no. 12, p. 2985, 2024, https://doi.org/10.3390/en17122985.

[17] A. Chellakhi, S. El Beid, and Y. Abouelmahjoub, “An improved adaptable step-size P&O MPPT approach for standalone photovoltaic systems with battery station,” *Simulation Modelling Practice and Theory*, vol. 121, p. 102655, 2022, https://doi.org/10.1016/j.simpat.2022.102655.

[18] C. Aoughlis, A. Belkaid, I. Colak, O. Guenounou and M. A. Kacimi, "Automatic and Self Adaptive P&O MPPT Based PID Controller and PSO Algorithm," *2021 10th International Conference on Renewable Energy Research and Application (ICRERA)*, pp. 385-390, 2021, https://doi.org/10.1109/icrera52334.2021.9598489.

[19] R. N. Hasanah, F. Yuniar, O. Setyawati, H. Suyono, D. R. Sawitri, and T. Taufik, “A Modified Perturb-and-Observe Control for Improved Maximum Power Point Tracking Performance on Grid-Connected Photovoltaic System,” *International Journal of Technology*, vol. 15, no. 1, pp. 99-109, 2024, https://doi.org/10.14716/ijtech.v15i1.5316.

[20] A. Chellakhi, S. El Beid, Y. Abouelmahjoub, and H. Doubabi, “An Enhanced Incremental Conductance MPPT Approach for PV Power Optimization: A Simulation and Experimental Study,” *Arabian Journal for Science and Engineering*, vol. 49, no. 12, pp. 16045- 16064, 2024, https://doi.org/10.1007/s13369-024-08804-1.

[21] S. F. Chevtchenko, E. J. Barbosa, M. C. Cavalcanti, G. M. S. Azevedo, and T. B. Ludermir, “Combining PPO and incremental conductance for MPPT under dynamic shading and temperature,” *Applied Soft Computing*, vol. 131, p. 109748, 2022, https://doi.org/10.1016/j.asoc.2022.109748.

[22] S. Rafi Kiran and F. Alsaif, “A novel advanced hybrid fuzzy MPPT controllers for renewable energy systems,” *Scientific Reports*, vol. 14, no. 1, pp. 1-17, 2024, https://doi.org/10.1038/s41598-024-72060-4.

[23] K. Ananda-Rao *et al*., “MPPT Charge Controller using Fuzzy Logic for Battery Integrated with Solar Photovoltaic System,” *Journal of Advanced Research in Applied Sciences and Engineering Technology*, vol. 47, no. 2, pp. 171-182, 2025, https://doi.org/10.37934/ARASET.47.2.171182.

[24] S. S. Kumar and K. Balakrishna, “A novel design and analysis of hybrid fuzzy logic MPPT controller for solar PV system under partial shading conditions,” *Scientific Reports*, vol. 14, no. 1, pp. 1-17, 2024, https://doi.org/10.1038/s41598-024-60870-5.

[25] K. Krishnaram, T. S. Padmanabhan, F. Alsaif, and S. Senthilkumar, “Performance optimization of interleaved boost converter with ANN supported adaptable stepped-scaled P&O based MPPT for solar powered applications,” *Scientific Reports*, vol. 14, no. 1, pp. 1-17, 2024, https://doi.org/10.1038/s41598-024-58852-8.

[26] P. R. Bana, S. D’Arco, and M. Amin, “ANN-Based Robust Current Controller for Single-Stage Grid-Connected PV With Embedded Improved MPPT Scheme,” *IEEE Access*, vol. 12, pp. 100251–100262, 2024, https://doi.org/10.1109/ACCESS.2024.3429347.

[27] M. Mishra, P. Mahajan, and R. Garg, “Implementation and comparison of metaheuristically modified ANN MPPT controllers under varying solar irradiance conditions,” *Electrical Engineering*, vol. 106, no. 3, pp. 3427-3443, 2024, https://doi.org/10.1007/s00202-023-02165-y.

[28] H. Abouobaida, Y. Mchaouar, Y. Abouelmahjoub, H. Mahmoudi, A. Abbou, and M. Jamil, “Performance optimization of the INC-COND fuzzy MPPT based on a variable step for photovoltaic systems,” *Optik*, vol. 278, p. 170657, 2023, https://doi.org/10.1016/j.ijleo.2023.170657.

[29] M. J. Alshareef, "An Enhanced Fractional Open Circuit Voltage MPPT Method for Rapid and Precise MPP Tracking in Standalone Photovoltaic Systems," *IEEE Access*, vol. 13, pp. 34115-34131, 2025, https://doi.org/10.1109/ACCESS.2025.3543327.

[30] A. B. Tadesse, E. A. Ayele, and A. O. Olonje, “Design and Analysis of Rate Predictive Fractional-Order Sliding Mode Controller (RP-FOSMC) for MPPT and Power Regulation of DFIG-based Wind Energy

Conversion System (WECS)," *Energy Reports*, vol. 8, pp. 11751-11768, 2022, https://doi.org/10.1016/j.egyr.2022.09.026.

[31] T. M. Shami, A. A. El-Saleh, M. Alswaitti, Q. Al-Tashi, M. A. Summakieh, and S. Mirjalili, "Particle Swarm Optimization: A Comprehensive Survey," *IEEE Access*, vol. 10, pp. 10031-10061, 2022, https://doi.org/10.1109/ACCESS.2022.3142859.

[32] I. Saady *et al*., "Improving photovoltaic water pumping system performance with PSO-based MPPT and PSO-based direct torque control using real-time simulation," *Scientific Reports*, vol. 15, no. 1, pp. 1-24, 2025, https://doi.org/10.1038/s41598-025-00297-8.

[33] M. J. Hasan and M. Y. Hellan, "Control and Modeling of PSO-PID based MPPT," *International Journal of Computer Applications*, vol. 184, no. 13, pp. 1-8, 2022, https://doi.org/10.5120/ijca2022922112.

[34] M. M. Elymany, M. A. Enany, and N. A. Elsonbaty, "Hybrid optimized-ANFIS based MPPT for hybrid microgrid using zebra optimization algorithm and artificial gorilla troops optimizer," *Energy Conversion and Management*, vol. 299, p. 117809, 2024, https://doi.org/10.1016/j.enconman.2023.117809.

[35] D. Mazumdar, P. K. Biswas, C. Sain, F. Ahmad, and L. Al-Fagih, "A comprehensive analysis of the optimal GWO based FOPID MPPT controller for grid-tied photovoltaics system under atmospheric uncertainty," *Energy Reports*, vol. 12, pp. 1921-1935, 2024, https://doi.org/https://doi.org/10.1016/j.egyr.2024.08.013.

[36] F. S. Pai and P. S. Tseng, "An efficient GWO MPPT for a PV system using impedance information acceleration," *International Journal of Electronics*, vol. 106, no. 4, pp. 648-661, 2019, https://doi.org/10.1080/00207217.2018.1545929.

[37] J. Li, S. Lu, and J. Yang, "Multi-Stage Cooperative Optimization Control for Photovoltaic MPPT: A High-Efficiency Gray Wolf Optimizer–Incremental Conductance Hybrid Strategy," *Energies*, vol. 18, no. 8, p. 1977, 2025, https://doi.org/10.3390/en18081977.

[38] K. Xia, Y. Li, and B. Zhu, "Improved Photovoltaic MPPT Algorithm Based on Ant Colony Optimization and Fuzzy Logic Under Conditions of Partial Shading," *IEEE Access*, vol. 12, pp. 44817-44825, 2024, https://doi.org/10.1109/ACCESS.2024.3381345.

[39] K. Kayisli, "Super twisting sliding mode-type 2 fuzzy MPPT control of solar PV system with parameter optimization under variable irradiance conditions," *Ain Shams Engineering Journal*, vol. 14, no. 1, p. 101950, 2023, https://doi.org/10.1016/j.asej.2022.101950.

[40] L. gang Kong, B. Wang, D. jin Fan, S. Shi, X. Ouyang, and M. Xu, "Optimize photovoltaic MPPT with improved snake algorithm," *Energy Reports*, vol. 11, pp. 5033-5045, 2024, https://doi.org/10.1016/j.egyr.2024.04.064.

[41] M. K. Senapati, C. Pradhan, S. Padmanaban, and O. Al Zaabi, "Photovoltaic MPPT Performance Adaptability to Partial Shading Resilience and Load Variations With Modified Adaptive Jaya Optimization," *IEEE Transactions on Consumer Electronics*, vol. 71, no. 1, pp. 734-747, 2025, https://doi.org/10.1109/TCE.2025.3532660.

[42] S. Nurcahyo, H. Suyono, R. N. Hasanah, and M. A. Muslim, "Integration of PID-MRAC and Novel GCC-C2C for Developing Adaptive Deterministic MPPT," *Journal of Robotics and Control (JRC)*, vol. 6, no. 4, pp. 1636-1647, 2025, https://doi.org/10.18196/JRC.V6I4.26794.

[43] M. S. Bakare, A. Abdulkarim, A. N. Shuaibu, and M. M. Muhamad, "Enhancing solar power efficiency with hybrid GEP ANFIS MPPT under dynamic weather conditions," *Scientific Reports*, vol. 15, no. 1, p. 5890, 2025, https://doi.org/10.1038/s41598-025-90417-1.

[44] R. Bisht, A. Sikander, A. Sharma, K. Abidi, M. R. Saifuddin, and S. S. Lee, "A New Hybrid Framework for the MPPT of Solar PV Systems Under Partial Shaded Scenarios," *Sustainability*, vol. 17, no. 12, p. 5285, 2025, https://doi.org/10.3390/SU17125285.

[45] A. F. Murtaza, A. Hussain, H. A. Sher, A. Alsaleem, and F. Spertino, "A Simple Integration Architecture of Photovoltaic Plant for Consumer Electronics in LVDC Systems," *IEEE Transactions on Consumer Electronics*, vol. 71, no. 2, pp. 2689-2701, 2025, https://doi.org/10.1109/TCE.2025.3574805.

[46] K. A. Bhagwan, U. B. Manthati, and F. Alsaif, "A model predictive control based MPPT technique for novel DC-DC converter and voltage regulation in DC microgrid," *Frontiers in Energy Research*, vol. 12, p. 1471499, 2024, https://doi.org/10.3389/fenrg.2024.1471499.

[47] B. Nishanthi and J. Kanakaraj, "Voltage Stability Assessment and Power Regulation of Solar PV Based DC Microgrid," *Journal of Electrical Engineering and Technology*, vol. 20, no. 1, pp. 131-140, 2025, https://doi.org/10.1007/S42835-024-02013-4/FIGURES/20.

[48] I. Bekki, H. Rizki, F. Lamzouri, A. El Amrani, and E. M. Boufounas, "Enhanced MPPT and voltage regulation in PV–electrolyzer systems using backstepping integral sliding mode control," *Electrical Engineering*, vol. 107, pp. 13737-13754, 2025, https://doi.org/10.1007/s00202-025-03232-2.

[49] M. Golla, D. V. Sudarsan Reddy, and S. Thangavel, "A unified single-controller approach of an MPPT and droop control for solar PVs in DC microgrid under unequal and partial shading conditions," *Renewable Energy*, vol. 256, p. 123940, 2026, https://doi.org/10.1016/J.RENENE.2025.123940.

[50] A. Ullah, S. S. Rizvi, A. Khatoon, and S. Jin Kwon, "The Empirical Analysis, Mathematical Modeling, and Advanced Control Strategies for Buck Converter," *IEEE Access*, vol. 12, pp. 19924-19941, 2024, https://doi.org/10.1109/ACCESS.2024.3357537.

[51] S. M. Ghamari, H. Mollaee, and F. Khavari, "Robust self-tuning regressive adaptive controller design for a DC–DC BUCK converter," *Measurement*, vol. 174, p. 109071, 2021, https://doi.org/10.1016/j.measurement.2021.109071.

[52] L. K. Fong, M. S. Islam, and M. A. Ahmad, "Optimized PID Controller of DC-DC Buck Converter based on Archimedes Optimization Algorithm," *International Journal of Robotics and Control Systems*, vol. 3, no. 4, pp. 658-672, 2023, https://doi.org/10.31763/ijrcs.v3i4.1113.

[53] W. Zhang *et al*., "Integrated irrigation of water and fertilizer with superior self-correcting fuzzy PID control system," *PLoS One*, vol. 20, no. 5, p. e0324448, 2025, https://doi.org/10.1371/JOURNAL.PONE.0324448.

[54] N. W. Madebo, "Enhancing Intelligent Control Strategies for UAVs: A Comparative Analysis of Fuzzy Logic, Fuzzy PID, and GA-Optimized Fuzzy PID Controllers," *IEEE Access*, vol. 13, pp. 16548-16563, 2025, https://doi.org/10.1109/ACCESS.2025.3532743.

[55] N. Liu, T. Chai, Y. Zhang, and W. Gao, "Data-driven optimal tuning of PID controller parameters," *Science China Information Sciences*, vol. 68, no. 7, p. 172201, 2025, https://doi.org/10.1007/s11432-024-4292-y.

[56] J. C. Almachi, R. Vicente, E. Bone, J. Montenegro, E. Cando, and S. Reina, "Implementation of a Neural Network for Adaptive PID Tuning in a High-Temperature Thermal System," *Energies*, vol. 18, no. 12, p. 3113, 2025, https://doi.org/10.3390/EN18123113.

[57] Y. O. M. Sekyere, P. O. Ajiboye, F. B. Effah, and B. T. Opoku, "Optimizing PID control for automatic voltage regulators using ADIWACO PSO," *Scientific African*, vol. 27, p. e02562, 2025, https://doi.org/10.1016/J.SCIAF.2025.E02562.

[58] J. A. van Niekerk, J. D. le Roux, and I. K. Craig, "Reinforcement learning based automatic tuning of PID controllers in multivariable grinding mill circuits," *Control Engineering Practice*, vol. 165, p. 106522, 2025, https://doi.org/10.1016/j.conengprac.2025.106522.

[59] N. P. Lawrence, M. G. Forbes, P. D. Loewen, D. G. McClement, J. U. Backström, and R. B. Gopaluni, "Deep reinforcement learning with shallow controllers: An experimental application to PID tuning," *Control Engineering Practice*, vol. 121, p. 105046, 2022, https://doi.org/10.1016/j.conengprac.2021.105046.

[60] M. Caparroz, K. Soltesz, T. Hägglund, J. L. Guzmán, and M. Berenguel, "A new approach to relay-based autotuning PID controllers and their evaluation in pH control of industrial photobioreactors," *Control Engineering Practice*, vol. 164, p. 106520, 2025, https://doi.org/10.1016/j.conengprac.2025.106520.

[61] S. Nurcahyo, F. Fitri, and S. Sungkono, "Development of a Digital Autotuning PI for First Order Plant Using RLS-PZC," *Journal of Robotics and Control (JRC)*, vol. 6, no. 1, pp. 12-21, 2025, https://doi.org/10.18196/jrc.v6i1.24257.

[62] T. Khan and C. Mahanta, "An intelligent adaptive control of DC – DC buck converters," *Journal of the Franklin Institute*, vol. 353, no. 12, pp. 2588-2613, 2016, https://doi.org/10.1016/j.jfranklin.2016.04.008.

[63] I. AL-Wesabi *et al*., "Hybrid SSA-PSO based intelligent direct sliding-mode control for extracting maximum photovoltaic output power and regulating the DC-bus voltage," *International Journal of Hydrogen Energy*, vol. 51, pp. 348-370, 2024, https://doi.org/10.1016/j.ijhydene.2023.10.034.

[64] L. T. Rasheed, "Bat Algorithm Based an Adaptive PID Controller Design for Buck Converter Model," *Journal of Engineering*, vol. 26, no. 7, pp. 62-82, 2020, https://doi.org/10.31026/j.eng.2020.07.05.

[65] P. Warrier and P. Shah, "Optimal fractional pid controller for buck converter using cohort intelligent algorithm*," Applied System Innovation*, vol. 4, no. 3, pp. 1-21, 2021, https://doi.org/10.3390/asi4030050.

[66] J. Kim, F. Y. Lee, J. Lee, and J. S. Il Kwon, "Nonlinear second order plus time delay model identification and nonlinear PID controller tuning based on extended linearization method," *Control Engineering Practice*, vol. 152, p. 106044, 2024, https://doi.org/10.1016/j.conengprac.2024.106044.

[67] S. Prajapati, R. Garg, and P. Mahajan, "Modified control approach for MPP tracking and DC bus voltage regulation in a hybrid standalone microgrid," *Electric Power Systems Research*, vol. 236, p. 110935, 2024, https://doi.org/10.1016/j.epsr.2024.110935.

[68] C. M. Aung, E. Mon, T. T. Htike, and H. M. Tun, "Comparative Analysis of Arduino-microcontroller based on PWM and MPPT Solar Charge Controller for Stand-alone Solar PV System in Charging Stations of Electric Vehicles," *Engineering Science Letter*, vol. 4, no. 02, pp. 32-38, 2025, https://doi.org/10.56741/IISTR.esl.001006.

[69] M. Bathre and P. K. Das, "Design & implementation of smart power management system for self-powered wireless sensor nodes based on fuzzy logic controller using Proteus & Arduino Mega 2560 microcontroller," *Journal of Energy Storage*, vol. 97, p. 112961, 2024, https://doi.org/10.1016/J.EST.2024.112961.

[70] A. A. Azi, D. Saigaa, M. Drif, and A. Loukriz, "Development of generalized photovoltaic model using ISIS-PROTEUS," *Studies in Engineering and Exact Sciences*, vol. 5, no. 2, p. e5546, 2024, https://doi.org/10.54021/seesv5n2-015.

[71] M. R. Mojallizadeh *et al*., "Control design for thrust generators with application to wind turbine wave-tank testing: A sliding-mode control approach with Euler backward time-discretization," *Control Engineering Practice*, vol. 146, p. 105894, 2024, https://doi.org/10.1016/j.conengprac.2024.105894.

[72] T. Abdelkarim, R. Driss, O. Eichwald, L. Vido, N. Kamel and B. Abdelkrim, "Influence of the Forgetting Factor in the Recursive Least Squares RLS Algorithm on the Quality and Precision of the Identified Parameters in a DC Corona Discharge," *IEEE Transactions on Plasma Science*, vol. 53, no. 1, pp. 108-115, 2025, https://doi.org/10.1109/TPS.2024.3524470.

[73] X. Ning, Z. Wang, C. Wang, and B. Wu, "Adaptive Feedforward and Feedback Compensation Method for Real-time Hybrid Simulation Based on a Discrete Physical Testing System Model," *Journal of Earthquake Engineering*, vol. 26, no. 8, pp. 3841-3863, 2022, https://doi.org/10.1080/13632469.2020.1823912.