PID Parameter Tuning for Buck Controllers Based on an Improved Differential Evolution Algorithm

YJ Zhao¹,a *, QX Qiu¹,a, X Zhao¹,a

¹Zhengzhou Electromechanical Engineering Research Institute, ZhengZhou, Henan, China
²email: zhaoyingjiecai@cssc713.com,

Abstract: With the wide application of Buck converters, it is of great significance to select the optimal PID controller parameters to ensure the dynamic and steady-state performance of the Buck converters. On this basis, a PID parameter tuning method based on an improved differential evolution algorithm is proposed. Also, a crossover approach with population generation across orders of magnitude, elastic boundary absorption, and retention of some combined features is put forward. In addition, an adaptive mutation factor, and a cost function that suppresses overshoot and is sensitive to the tuning time are designed. The results of 10 tests show that the differential evolution algorithm fails in the case of parameters in a range of orders of magnitude, while the algorithm proposed in this paper enables a better combination of PID parameters selected in terms of speed and stability, with a wide search space and stable convergence results.

1. Introduction

With the small current ripple, simple structure, small size, low cost, and other advantages, Buck converters are widely used in switching power supplies, voltage regulators, charging circuits of lithium battery storage systems, supercapacitor energy storage systems, etc. However, the Buck converters are time-varying nonlinear systems, which have the disadvantage of poor anti-interference ability. Given this, a suitable controller needs to be introduced to make the Buck converters control the targeted output voltage in a fast and stable way by pulse-width modulation (PWM) [1].

PID control is the most widely used control method in the industry. Due to its advantages of a simple algorithm, good robustness, and high reliability, it is widely used in power electronics and instrumentation. Most of the current PID controller parameters are adjusted using engineering tuning methods, such as the rule-based method represented by the Ziegler–Nichols (Z-N) tuning method and the model-based method represented by the gain and phase margin (GPM) method. The tuning of these parameters requires more work volume and relies more on the experience of technicians, and the results are often not optimized enough in the case of high-performance indicator requirements [2]. The optimization of PID controller parameters has become a concern, which directly affects the control effect and is closely related to the safety and economic operation of the system.

There are many methods of PID parameter optimization, such as gradient method, hill-climbing method, simplex method, but they are easy to form a local optimum and have high requirements for the optimized object. At present, intelligent algorithms also stand out in parameter optimization, and numerous numerical optimization analysis examples have verified that the differential evolution algorithm [3] has better optimization and convergence performance than the genetic algorithm [4] and...
particle swarm algorithm [5][6]. To further improve the optimization and convergence performance of the differential evolution (DE) algorithm and to adapt to the actual situation of a large range of PID parameters for tuning, this paper modifies the generation strategy of initial population, optimizes the boundary absorption scheme, introduces an adaptive scaling factor, designs a more reasonable PID fitness function, and then applies it to the optimal tuning of PID parameters of the Buck control circuit.

2. Modeling of A Buck Converter

The Buck converter is a single-tube non-isolated DC converter, whose topology is shown in Figure 1. The voltage control Buck converter consists of a power circuit and a control circuit. The power circuit is to convert the input voltage to the voltage required by the load, and it contains the power switching tube Q, flyback diode D, inductor L, capacitor C, and load R. The control circuit mainly comprises a regulator and voltage dividers R4, R5 and PWM. When the external disturbance occurs, the difference between the sampled voltage and the reference input voltage Vref enables PWM control through the control network or digital controller, so as to get a suitable control signal to turn on or off the power switching tube Q, ensuring stable output of the power circuit.

Due to the presence of nonlinear components such as power switching tubes and diodes, the Buck converter is a nonlinear system. But when the operating characteristics near a stable operating point are studied, the converter can be approximated as a linear system, and the Buck equivalent circuit under the small signal model is shown in Figure 2.

According to the continuous conduction condition of the Buck converter, an inductor shall be selected.

$$L = \frac{V_o D(1-D)}{2f_s I_{oc}} \quad (1)$$

Where $I_{oc} = 0.2I_o$, the duty cycle D is 2, the output voltage ripple factor is required to be 1%, i.e., the pulse value of the output voltage, and $\Delta V_o \leq 0.01V_o$. From the relationship between the filter capacitor charge and discharge, it can be obtained that:

$$\Delta V_o = \frac{(1-D)V_o}{8LCf_s^2} \quad (2)$$

Based on the final comprehensive consideration, a parameter list is obtained, as shown in Table 1.
A linear model is established using the small-signal perturbation method. Each state variable and input current in the Buck converter circuit changes slightly due to small-signal perturbation, which can be expressed as the sum of the DC component and the AC small-signal component. By discarding the second-order AC term and retaining the DC term and the first-order AC term, the transfer function of the s-domain circuit of the Buck converter can be obtained as follows.

\[ sL \frac{\hat{L}(s)}{L} = D \frac{\hat{v}_g(s)}{V_g} \frac{\hat{d}(s)}{d(s)} - \hat{v}(s) \]

\[ sC \hat{\nu}(s) = \hat{I}_L(s) - \frac{\hat{v}(s)}{R} \]

\[ \hat{i}_g(s) = D \frac{\hat{i}_L(s)}{I_g} + \frac{I_d(s)}{d(s)} \]

The control-output transfer function \( G_{vd}(s) \) is as follows.

\[ G_{vd}(s) = \frac{\hat{v}(s)}{d(s)} \bigg|_{\hat{d}(s)=0} = \frac{V_g}{LCs^2 + \frac{L}{R} s + 1} \]

According to the block diagram of the Buck converter control structure in Figure 3, we can obtain that:

\[ \hat{v}(s) = G_{vd}(s) \hat{d}(s) + G_{vg}(s) \hat{v}_g(s) + Z_o(s) \hat{i}_g(s) \]
The data in Table 1 can be substituted to obtain the original open-loop transfer function.

\[
G_o(s) = G_m(s)G_{vd}(s)H(s) = \frac{HV_m}{LCs^2 + \frac{L}{R}s + 1} = \frac{1280}{8 \times 10^{-8}s^2 + 0.001s + 8} \tag{7}
\]

The open-loop loop Bode plot of Eq. (7) is shown in Fig. 4. The DC gain is 44.3 dB and is limited with static error. The gain crossover frequency \( w_g = 1.27 \times 10^5 \text{rad/s} \), and the phase margin PM = 5.68°, which is too small, causing oscillations in the system output.

![Fig. 4 Loop Gain Bode Plot of Buck Converter](image)

### 3. PID Parameter Tuning Based on Improved Differential Evolution Algorithm

The differential evolution algorithm retains the population-based global search strategy and uses real number encoding, simple difference-based mutation operation, and one-to-one competitive survival strategy, reducing the complexity of the genetic operation. The unique memory capability of the differential evolution algorithm allows it to dynamically track the current search situation to adjust its search strategy. Also, due to its strong global convergence and robustness, without the need to resort to the feature information of problems, the algorithm is best suited for solving some nonlinear, non-differential, multi-extremum, and high-dimensional complex problems. In recent years, differential evolution has been increasingly used in control applications. \( T_n \) is the sampling time. Then the positional discrete PID expression is as follows:

\[
u(k) = k_p e(k) + \frac{k_i}{T_n} (e(k) - e(k-1)) + k_d \sum_{j=1}^{4} e(k)T_n^j \tag{8}\]

The discretization of Eq. (7) is performed with the zero-order hold (ZOH) method. The sampling time is taken as 1 \( \mu s \). The specific steps for optimizing \( k_p \), \( k_i \), and \( k_d \) using differential evolution are as follows.

**Step1: Initialization.** The population is generally taken as 10 times the dimensionality, so 30 individuals satisfying the constraints are randomly generated in the three-dimensional space. The initial population is \( S = \{X_1, X_2, \ldots, X_{30}\} \) and the \( i \) th individual is \( X_i = \{x_{i1}, x_{i2}, x_{i3}\} \). Since the PID parameters are not in the same order of magnitude, the initial population is randomly generated by the logarithmic method as follows.

\[
x_i(0) = \begin{cases} \text{rand}_{ij}(0,1)(x_{ij}^U - x_{ij}^L) + x_{ij}^L, & x_{ij}^U / x_{ij}^L < 100 \\ 10^{\left(\frac{x_{ij}^U - x_{ij}^L}{\log x_{ij}^U - \log x_{ij}^L}\right)} \text{rand}_{ij}(0,1) + \log x_{ij}^U, & x_{ij}^U / x_{ij}^L \geq 100 \end{cases} \tag{9}\]

Where \( \text{rand}_{ij}(0,1) \) is the random number between 0 and 1, and \( x_{ij}^U \) and \( x_{ij}^L \) are the upper and lower limits of the individuals. In the early stage, according to the Z-N method combined with the human experience in the frequency domain, the parameters are tuned and the PID parameters are selected, i.e., the upper and lower limits of the individual values are shown in Table 2. And the lower
and upper limits are selected for better stability and speed, respectively. The unit step response of the upper and lower limit parameters is shown in Figure 5, and their performance indicators are shown in Table 3.

### Table 2 PID Parameter Range

| Parameter Name | Parameter lower Limit | Parameter upper Limit |
|----------------|------------------------|-----------------------|
| $k_p(x_1)$    | $3 \times 10^{-3}$     | $4 \times 10^{-2}$    |
| $k_i(x_2)$    | 35                     | 100                   |
| $k_d(x_3)$    | $9 \times 10^{-13}$    | $2 \times 10^{-6}$    |

![Step Response](image1.png)  
**Fig. 5** Step Response of Upper and Lower Limit Parameters

### Table 3 Performance Indicators of Upper and Lower Limit Parameters

| Parameter Category | OverShoot | Rise Time | Setting Time | Steady State | Number of oscillations |
|--------------------|-----------|-----------|--------------|--------------|------------------------|
| Upper limit        | 28.7%     | 0.33ms    | 1.8ms        | 1            | 2                      |
| Lower limit        | 0%        | 0.054ms   | 0.77ms       | 1            | 1                      |

Step2: Mutation operation. Three individuals, which are $X_b$, $X_n$, and $X_w$ ($i \neq b \neq n \neq w$), are randomly selected from the current population, and their cost function values increase in order. Then the mutation operation is as follows.

$$h_i(t + 1) = X_{b_i}(t) + F(X_{r_3}(t) - X_{r_2}(t))$$  \hspace{1cm} (10)$$

$X_{b_i}(t)$ is the best individual in the current generation, $X_{r_3}, X_{r_1}$ are two of the random factors, and $F$ is the adaptive scaling factor.

$$F = 0.1 + 0.8\left(\frac{f(X_n) - f(X_b)}{f(X_n) - f(X_w)}\right)$$  \hspace{1cm} (11)$$

Where $f(X_i)$ is the objective function of optimization. According to the requirements of regulating the time period without overshoot as much as possible, an integrated cost function is designed as follows.

$$f(X_i) = t_s \int_0^{nT_e} [|e(t)| + u^2(t)](1 + ([y(t) > 1]*500) + ([y(t) < 0]*5000))dt$$  \hspace{1cm} (12)$$

Where $t_s$ is the regulation time (within a stable range of 5%), $n$ is the calculation step, $T_e$ is the
sampling time, \(e(t)\) is the error, \(u(t)\) is the controller output. A 500-fold penalty is given for overshoot and a 5000-fold penalty is given for an unstable range. If the generated \(h_y(t + 1)\) is beyond the boundary, to prevent falling into the boundary trap, the optimized elastic boundary absorption function is added.

\[
h_y(t + 1) = \begin{cases} x_y^L + \text{rand}_y(0,0.1)x_y^L, & h_y(t + 1) < x_y^L \\ x_y^U - \text{rand}_y(0,0.1)x_y^L, & h_y(t + 1) > x_y^U \end{cases}
\]

(13)

Step3: Crossover operation. Since the feature combination of \(k_p\), \(k_i\), and \(k_d\) determines the control performance, we need to preserve the combined features of the solution as much as possible. Let \(\text{rand}(1,2,3)\) be two randomly generated integers between 1 and 3, and \(v_y(t + 1)\) be the test vector. Then a two-dimensional crossover operation can be designed as follows.

\[
v_y(t + 1) = \begin{cases} x_y(t), & j \notin \text{rand}(1,2,3) \\ h_y(t + 1), & j \in \text{rand}(1,2,3) \end{cases}
\]

(14)

Step4: Selection operation. To determine whether \(v_y(t + 1)\) and \(x_y(t)\) can be members of the next generation, their cost functions are compared as follows.

\[
x_y(t + 1) = \begin{cases} x_y(t), & f(X_j) \geq f(V_j) \\ v_y(t + 1), & f(X_j) < f(V_j) \end{cases}
\]

(15)

Repeat Step 2 to Step 4 until the maximum evolutionary generation \(G\) is reached.

4. Results&Discussion

This paper ran the unimproved differential evolution algorithm 10 times under the upper and lower limit parameters in Table 2. The results obtained were basically consistent, with problems such as small search range, easy convergence on the boundary, limited population mutation, and premature. The PID parameters basically cannot be optimized properly and the convergence results of the DE algorithm are shown in Fig. 6.

![Convergence Process Diagram of Differential Evolution Algorithm](image)

Fig. 6 Convergence Process Diagram of Differential Evolution Algorithm

With the data in Table 2 as the upper and lower limit parameters, the differential evolution algorithm is improved using Eq. (9) to Eq. (15) to optimize the PID parameter tuning 10 times. 10 evolutionary iterations were conducted for each experiment. During the experiment, the solution-solving method for the tuning time could be modified as needed, and the tuning time in this experiment remained stable within 0.1%. The experimental results are shown in Table 4, where the results of the 4th run are shown in Fig. 7. Compared with Fig. 6, the convergence process achieves the optimization purpose. The process converges in the 32nd generation, and finally converges between 0.03 and 0.05, and the cost function reduces by double. By doing so, the good individuals are preserved and reproduced. Compared with Fig. 5, the optimized results not only integrate the
advantages of upper and lower limit parameters but also improve the dynamic performance. No overshoot and oscillation occur during the 10 experiments, and the tuning time reduces from 0.77 ms to 0.28 ms. This result shows that the improved differential evolution algorithm proposed in this paper has superior performance in PID parameter optimization.

| No. | $K_p$   | $K_d$   | $K_i$   | tr/s    | ts/s    | F(x)    |
|-----|---------|---------|---------|---------|---------|---------|
| 1   | 0.0121624 | 9.62983e-07 | 97.7746 | 0.000145 | 0.000242 | 0.0351168 |
| 2   | 0.00773121 | 6.19433e-07 | 65.5785 | 0.000217 | 0.000335 | 0.0489274 |
| 3   | 0.0074018 | 6.02241e-07 | 62.9238 | 0.000228 | 0.000352 | 0.0526359 |
| 4   | 0.0110659 | 8.78301e-07 | 91.1874 | 0.000157 | 0.00025  | 0.0276536 |
| 5   | 0.0095317 | 7.67792e-07 | 79.2239 | 0.000181 | 0.000289 | 0.0408831 |
| 6   | 0.00790785 | 6.373e-07 | 66.8966 | 0.000214 | 0.000332 | 0.0468284 |
| 7   | 0.0103944 | 8.27318e-07 | 85.3515 | 0.000167 | 0.00027  | 0.0330632 |
| 8   | 0.00936562 | 7.41612e-07 | 77.5672 | 0.000183 | 0.000291 | 0.0398157 |
| 9   | 0.0101521 | 8.02542e-07 | 84.093  | 0.000169 | 0.000267 | 0.0321795 |
| 10  | 0.00757473 | 6.23893e-07 | 62.7639 | 0.000231 | 0.000373 | 0.0563088 |

5. **Conclusions**

To improve the dynamic output performance of the Buck converter, this paper first established an accurate Buck converter transfer function using the small signal model, and then designed two sets of PID parameters using the empirical method, but their performance indicators were not ideal. In order to get the PID parameters with better dynamic performance, this paper proposed an improved differential evolution algorithm that can effectively optimize PID parameters and improve the system's dynamic performance.
algorithm. To adapt to the characteristics of the PID parameters, a crossover approach with population generation across orders of magnitude, elastic boundary absorption, and retention of some combined features was put forward. Besides, a mutation strategy based on the adaptive degree function was introduced. The results show that the PID self-tuning method proposed in this paper can reduce the regulation time from 0.77 ms to 0.25 ms and eliminate overshoot and oscillation.

Although PID control is a mature and widely used control method, the design of its parameters has a great impact on the system. However, limited by experience and practical engineering problems, the method is often difficult to obtain the best controller parameters. Considering the characteristics of PID parameters, the improved differential evolution algorithm proposed in this paper has practical engineering significance in PID parameter optimization.

References
[1] Guiza, D., Ounnas, D., Soufi, Y., & Bouden, A. (2021). DC-DC Buck Converter Control Improvement. 2021 18th International Multi-Conference on Systems, Signals & Devices (SSD).
[2] Somefun O A, Akingbade K, Dahunsi F. (2021). The dilemma of PID tuning. Annual Reviews in Control.
[3] Storn, R., & Price, K. (1997). Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization, 11(4), 341-359.
[4] Priya, K. M., & Balakumaran, T. (2021). GA based PID parameters optimization for electric vehicle application. International Conference on Computational Sciences-modelling, Computing and Soft Computing (CSMCS 2020).
[5] Su, S., Zhai, Z., Wang, C., & Ding, K. (2021). Improved fractional-order pso for pid tuning. International Journal of Pattern Recognition and Artificial Intelligence.
[6] Rahnamayan, S., Tizhoosh, H., Salama, & Ma, M. (2008). Opposition-based differential evolution. Evolutionary Computation IEEE Transactions on.