Power Electronics and Drives

# Optimal Tuning of Digital PID Controllers for Commercial BLDC Motors Using the Nelder-Mead Method

**Research Paper** 

Son T. Nguyen<sup>\*®</sup>, Tu M. Pham<sup>®</sup>, Anh Hoang<sup>®</sup>, Trung T. Cao<sup>®</sup>

School of Electrical and Electronic Engineering, Hanoi University of Science and Technology, 1 Dai Co Viet Street, Hanoi, Vietnam

Received: 15 April, 2025; Received in the revised form: 07 June, 2025; Accepted: 27 June, 2025

Abstract: Brushless DC (BLDC) motors are widely used in various applications due to their high efficiency and reliability. Many sensorless BLDC motor drivers, developed using zero-crossing detection of back electromotive force (EMF), eliminate the need for position sensors such as Hall sensors. The speed of a BLDC motor can be conveniently controlled using the trapezoidal control method, in which pulse-width modulation (PWM) is applied to the input DC voltage of the converter to regulate motor speed. Although Proportional-integral-derivative controller (PID) controllers are commonly used, accurately tuning their parameters remains difficult and challenging. Therefore, this study explores the use of the Nelder–Mead optimisation method for tuning digital PID controllers in commercial outer-rotor BLDC motors. Simulation and experimental results demonstrate that the Nelder-Mead method is highly effective for PID tuning in commercial BLDC motor drives, resulting in significant reductions in settling time and system overshoot.

Keywords: BLDC motor • PID controller • the Nelder-Method method

# 1. Introduction

The adoption of brushless DC (BLDC) motors has grown rapidly across a wide range of industries, including electric vehicles (EVs), industrial automation and home appliances. This growth can be attributed to their enhanced efficiency, durability and minimal maintenance requirements compared with traditional brushed DC motors. BLDC motors are particularly notable for their impressive energy efficiency, which optimises power usage, cuts down on energy consumption and helps reduce operating expenses. These advantages make them a top choice for applications that emphasise sustainability and long-term energy savings (Lee et al., 2016; Mohanraj et al., 2022).

In household appliances, BLDC motors offer quieter operation and improved efficiency in devices such as water pumps (Khan and Miah, 2020) and washing machines (Ho et al., 2012). Within the automotive sector, they enhance the performance, power density and torque of EVs, contributing to both higher performance and greater sustainability (Shao, 2006). Their lightweight design and reduced heat generation make them ideal for modern EV propulsion systems (Chau et al., 2007). Beyond consumer products, BLDC motors are extensively used in robotics (De et al., 2019) due to their precise motion control and high-speed capabilities. As technological advancements continue, developments in motor control, power electronics and battery technology are boosting both the efficiency and cost-effectiveness of BLDC motors, reinforcing their crucial role in the future of electrical and mechanical systems.

Since PID controller tuning methods based on the Ziegler-Nichols approach may not provide optimal performance for complex, non-linear systems like BLDC motors, several advanced techniques for optimal PID tuning have been proposed to address these limitations. The Ziegler-Nichols method, while effective for many

<sup>\*</sup> Email: son.nguyenthanh@hust.edu.vn

linear control systems, often struggles with highly dynamic and non-linear characteristics, such as those exhibited by BLDC motors (Bazanella et al., 2017). These limitations can lead to suboptimal transient response, overshoot and steady-state error, which negatively impact the overall performance and stability of the motor drive system. To overcome these challenges, more sophisticated and adaptive tuning approaches are required to ensure optimal performance across a wide range of operating conditions (Basilio and Matos, 2002; Huba et al., 2024).

Advanced PID controller tuning methods such as genetic algorithms (GA) (Chen and Perng, 2017), particle swarm optimisation (PSO) (Alberto et al., 2009), fuzzy logic (FL) (Gopan and Shree, 2022; Raja and Rathinakumar, 2023), Tabu search algorithm (Pakdeeto et al., 2023), artificial neural networks (ANN) (Owusu et al., 2023; Rubaai and Young, 2015) and model predictive control (MPC) (Ubare and Sonawane, 2022) have been proposed to enhance the performance of PID controllers in BLDC motor applications. These intelligent tuning techniques provide adaptive and self-optimising capabilities that allow the controller to dynamically adjust PID parameters in real time. As a result, they offer improved resilience and performance under varying operating conditions, such as fluctuating motor loads, changes in supply voltage, nonlinearities and system disturbances. By leveraging these methods, control systems can achieve faster response times, reduced steady-state error and better overall stability, which are critical for high-performance motor drive applications. Especially, swarm optimisation algorithms such as PSO have been proved to be powerful and versatile techniques for solving complex, high-dimensional and non-linear problems. They are also good at global search with local optima evasion at the conference of larger search space by their stochasticity, population-based and parallel nature.

One of the main challenges in BLDC motor drives is the presence of cogging torque causing undesirable vibrations, noise and fluctuations in speed, affecting the precision and smoothness of motor operation. Various strategies have been employed to mitigate cogging torque, one of which is DC-link voltage pulse amplitude modulation (PAM) (Huang et al., 2022). This technique effectively smooths the motor's torque output by modulating the voltage amplitude supplied to the motor drive, thereby reducing torque ripple and enhancing overall performance.

The development of sensorless control schemes for BLDC motors involves detecting an open-loop rotor position without physical position sensors such as Hall-effect sensors or encoders. Such a method allows cost reduction and system reliability enhancement and is also useful when sensors cannot be used. Sensorless control does away with physical sensors and estimates rotor position using methods like back-electromagnetic force (back-EMF) sensing, which works well at medium to high speeds but poorly at low speeds (Damodharan and Vasudevan, 2010); sliding mode observer (SMO) (Chen et al., 2000), which is robust but complex; extended Kalman filter (EKF) (Aishwarya and Jayanand, 2016), which is very accurate but computationally intensive; and flux linkage-based methods, which work well at low speeds (Gabriel and Nesimi, 2016). Furthermore, pulse-width modulation (PWM) strategies, including sine PWM (SPWM) with lower harmonics and space vector PWM (SVPWM) with higher efficiency, are also applied to control voltage and current (Gujjar and Kumar, 2017). The control method selection is application dependent; for basic applications, six-step commutation can be employed; for high-performance applications, field-oriented control (FOC)/direct torque control (DTC) could be chosen; and for cost-oriented solutions, sensorless control would be preferred (Bosso et al., 2021; Ozturk and Toliyat, 2011).

The Nelder–Mead optimisation method (Nelder and Mead, 1965) is one of the more effective alternatives to GAs and PSO for low-dimensional, smooth optimisation problems. One of its main advantages is the fast convergence without the use of gradients, therefore applicable to gradient-free optimisation problems. In addition, the Nelder–Mead use does not involve complicated stochastic processes like GA and PSO, as it uses a single simplex and fewer tuning parameters. This simplicity leads to less computational cost and more uniform and reproducible results. Therefore, it is common to use the Nelder–Mead method if the problem is unimodal and of low dimension and if fast and good local optimisation is required. The Nelder–Mead method, therefore, can be easily applied to tune the gains of a PID controller by minimising a chosen performance index. The method evolves a simplex of parameter sets by testing the control performance at every loop and adjusting PID parameters to be optimal expected behaviour without gradient information (Sinlapakun and Assawinchaichote, 2015).

Due to the substantial superiority of the Nelder–Mead method in comparison with other optimisation methods, this study proposes a systematic and precise algorithm for designing a digital PID controller for commercial sensorless BLDC motor drives. A PID parameter tuning using the Nelder–Mead optimisation algorithm is performed to minimise the weighted sum of a chosen performance index for obtaining better transient and steady-state responses of the BLDC motor drive. To verify and enhance the robustness of the Nelder–Mead optimisation algorithm for tuning PID controllers of BLDC motors, this study establishes a comprehensive and well-instrumented experimental platform. The hardware system is centred around a commercially available outer-

rotor brushless DC (BLDC) motor, chosen for its widespread use in industrial and consumer applications due to its high efficiency and compact design. Motor control is achieved using a commercial sensorless BLDC motor converter, which eliminates the need for physical position sensors by estimating rotor position through back-EMF, thereby reducing hardware complexity and cost.

The remaining parts of this paper are organised as follows: Section 2 explains the digital PID controller. The key steps of using the Nelder–Mead optimisation method for tuning PID controllers are presented in Section 3. Simulink-based simulation of two common control methods for BLDC motor drives, the trapezoidal control and FOC methods, is described in Section 4. The experimental setting and results are carefully mentioned in Section 5. Finally, Section 6 summarises the paper and future work is discussed.

# 2. The Digital PID Controller

The PID controller belongs to fundamental feedback control systems extensively utilised in automation, robotics and industrial process control. It dynamically determines the error by measuring the deviation between the target setpoint and the actual process variable. To minimise this error, the controller applies a calculated corrective response based on three key components as follows:

- · Proportional (P): Responds to the present error. A larger proportional gain leads to a more responsive system.
- Integral (I): Addresses the cumulative past errors. Although it helps eliminate steady-state errors, it may cause overshoot.
- Derivative (D): Responds to the rate of change of error. It anticipates future errors, helping to minimise overshoot
  and oscillations.

The output of a PID controller is calculated as follows:

$$u(t) = K_p e(t) + K_i \int e(t) dt + K_d \frac{de(t)}{dt}$$
<sup>(1)</sup>

Where:

u(t) is the control output,

e(t) is the error (setpoint-actual value),

 $K_p$  is the proportional gain,

 $K_i$  is the integral gain,

 $K_d$  is the derivative gain.

Taking the Laplace transform of Eq. (1) gives:

$$U(s) = K_p E(s) + \frac{K_i}{s} E(s) + K_d s E(s)$$
<sup>(2)</sup>

In Eq. (2), s is called the Laplace operator. Taking the z-transform of Eq. (2) results in:

$$U(z) = K_p E(z) + K_i \frac{zT}{z-1} E(z) + K_d \frac{z-1}{zT} E(z)$$
(3)

Where T is the sampling time. Eq. (3) is equivalent to the following expression:

$$U(z) = aE(z) + b\frac{1}{1-z^{-1}}E(z) + c\frac{1-z^{-1}}{1}E(z)$$
(4)

In which  $a = K_p$ ,  $b = K_i T$  and  $c = \frac{K_d}{T}$ . Eq. (4) can be shorted as follows:

$$U(z) = aE(z) + P(z) + Q(z)$$
<sup>(5)</sup>

Where:

$$P(z) = b \frac{1}{1 - z^{-1}} E(z)$$
(6)

$$Q(z) = c(1 - z^{-1})E(z)$$
<sup>(7)</sup>

Eq. (6) can be rewritten as follows:

$$P(z) = P(z)z^{-1} + bE(z)$$
(8)

To deploy digital PID controllers using digital hardware, the equations of the controller must be expressed in discrete-time forms as follows:

### 1) Calculate the error at the *k*-th sampling instant:

$$e_k = r_k - y_k \tag{9}$$

Where  $r_k$  is the setpoint and  $y_k$  is the output variable at the *k*-th sampling instant.

2) Calculate the integral term:	
$p_k = p_{k-1} + be_k$	(10)

3) Calculate the derivative term:

$$q_{k} = c(e_{k} - e_{k-1}) \tag{11}$$

Where  $e_k$  and  $e_{k-1}$  are the error at the k and (k-1)-th sampling intervals, respectively.

### 4) Calculate the control output:

$$u_k = ae_k + p_k + q_k \tag{12}$$

### 3. Tuning PID Controllers Using the Nelder–Mead Method

The Nelder–Mead method is a gradient-free optimisation method used to find the minimum of an objective function in a multi-dimensional space. It is particularly useful when derivatives of the objective function are unavailable or difficult to compute. This method is also known as the downhill simplex method. The deployment of the Nelder–Mead method includes the following steps:

### 1) Initialise the simplex:

Choose n+1 points in the search space. These points form a simplex, which is a geometric shape with n+1 vertices in an *n*-dimensional space.

### 2) Evaluate objective function:

Compute the function value at each simplex vertex.

### 3) Order the simplex:

Rank the points from best  $f(x_1)$  to worst  $f(x_{n+1})$  based on function values.

### 4) Apply transformations:

Update the worst point using one of the following transformations:

- Reflection: Reflect the worst point across the centroid of the remaining points.
- Expansion: If reflection improves the function significantly, expand further.
- Contraction: If reflection does not work well, try contracting towards the centroid.
- Shrink: If contraction fails, reduce the simplex size.
- 5) Repeat until convergence:

Continue iterating until a termination condition is met, such as:

(19)

- Function values have small differences.
- The simplex size becomes small.
- A maximum number of iterations is reached.

The details of mathematical operations in the Nelder–Mead method can be described as follows:

### Let:

- $x_1, x_2, \dots, x_{n+1}$  be the vertices of the simplex.
- $x_c$  is the centroid of the best *n* points

$$x_{c} = \frac{1}{n} \sum_{i=1}^{n} x_{i}$$
(13)

### Transformations:

1) Reflection:

$$x_r = x_c + \alpha \left( x_c - x_{n+1} \right) \tag{14}$$

2) Expansion:

$$x_e = x_c + \gamma \left( x_r - x_c \right) \tag{15}$$

If  $f(x_e)$  is better than  $f(x_r)$ , accept  $x_e$ .

### 3) Contraction:

Outside contraction:

 $x_{co} = x_c + \beta \left( x_r - x_c \right) \tag{16}$ 

Inside contraction:

$$x_{ci} = x_c - \beta (x_{n+1} - x_c) \tag{17}$$

• Shrink:  $x_i = x_1$ 

$$=x_1 + \delta(x_i - x_1), \quad i = 2, ..., n+1$$
 (18)

If none of the above works, shrink all the points towards the best.

Typical values of  $\alpha$ ,  $\gamma$ ,  $\beta$  and  $\delta$  can be set as follows:  $\alpha = 1$ ;  $\gamma = 2$ ;  $\beta = 0.5$  and  $\delta = 0.5$ .

The Nelder–Mead optimisation method can be used to optimally tune the PID controller's parameters, including  $K_p$ ,  $K_i$  and  $K_d$ . This process is carried out with the following steps:

### 1) Define the objective function:

The goal is to minimise an integral of time-weighted absolute error (ITAE) as follows:

$$J = |t|e(t)dt$$

2) Initialise the simplex:

Choose initial guesses for  $K_p$ ,  $K_i$  and  $K_d$ . Create a simplex with 3+1 (i.e., 4) sets of parameters.

### 3) Simulate the PID response:

- Apply the PID gains to the system using a step response.
- · Compute the objective function.

### 4) Apply Nelder–Mead transformations:

Adjust  $K_{p}$ ,  $K_{i}$  and  $K_{d}$  using reflection, expansion, contraction or shrinking.

### 5) Repeat until convergence:

Stop when a specified number of iterations is reached.

### 4. Control Simulation of BLDC Motors

### 4.1. Mathematical model of BLDC motors

A BLDC motor has three individual phase windings and a permanent magnet rotor. The three-phase voltage equations for windings are as follows:

$$\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = \begin{bmatrix} R_s & 0 & 0 \\ 0 & R_s & 0 \\ 0 & 0 & R_s \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} L_s - M & 0 & 0 \\ 0 & L_s - M & 0 \\ 0 & 0 & L_s - M \end{bmatrix} \frac{d}{dt} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} e_a \\ e_b \\ e_c \end{bmatrix}$$
(20)

In which  $v_a$ ,  $v_b$  and  $v_c$  are the phase voltages of the motor.  $R_s$  is the stator resistance and  $L_s$  is the stator self inductance. *M* is the mutual inductance between any two phases.  $i_a$ ,  $i_b$  and  $i_c$  are the stator currents. Finally,  $v_a$ ,  $v_b$  and  $v_c$  are back-EMF of the motor. The torque equation is given by:

$$T_e = \frac{e_a i_a + e_b i_b + e_c i_c}{\omega_m} \tag{21}$$

Here,  $\omega_m$  is the angular velocity of the motor and  $T_e$  is the electromagnetic torque. Since the back-EMF of the motor is trapezoidal, it can be obtained as follows:

$$\begin{bmatrix} e_a \\ e_b \\ e_c \end{bmatrix} = \omega_m \lambda_m \begin{bmatrix} f_{as}(\theta_r) \\ f_{bs}(\theta_r) \\ f_{cs}(\theta_r) \end{bmatrix}$$
(22)

Where  $\lambda_m$  is the flux leakage of the motor.  $f_{as}(\theta_r)$ ,  $f_{bs}(\theta_r)$  and  $f_{cs}(\theta_r)$  represent the functions of back-EMF for various magnitude instants. The motion equation of the motor is given by:

$$T_e - T_L = J \frac{d\omega_m}{dt} + F\omega_m \tag{23}$$

Here  $T_L$  is the load torque, J is the moment of inertia and F is the damping constant. Figure 1 shows the block diagram of a typical BLDC motor drive consisting of a three-phase voltage-source inverter and a BLDC motor.

BLDC motor control methods can be broadly categorised into sensor-based and sensorless techniques, each offering distinct advantages based on application requirements. Sensor-based control utilises Hall sensors or rotary encoders to accurately detect the rotor position, ensuring precise commutation and smooth motor operation. Among



Figure 1. Block diagram of a BLDC motor drive. BLDC, brushless DC

the most common sensor-based methods, trapezoidal control (six-step control) is widely used due to its simplicity and cost-effectiveness. FOC, on the other hand, employs mathematical transformations to regulate torque and flux independently, delivering smooth operation and high efficiency but requiring complex computations. The sensorless control of BLDC motors eliminates the need for physical sensors by leveraging back-EMF detection or advanced estimation algorithms, reducing cost and increasing reliability, though it may be less effective at low speeds. The selection of a suitable control method depends on factors such as cost, precision, efficiency and the operational conditions of the motor. This section presents simulations for controlling the speed of a BLDC motor using two common methods: the trapezoidal control method and the FOC method. The parameters of the BLDC motor are as follows:

- Stator phase resistance:  $R_s = 0.5(\Omega)$
- Stator phase inductance:  $L_s = 0.375 \times 10^{-3} (H)$
- Flux leakage of the motor:  $\lambda_m = 0.0080667 (V \cdot s)$
- Moment of inertia of the rotor:  $J = 3.989 \times 10^{-3} (\text{kg} \cdot \text{m}^2)$
- Damping constant:  $F = 0.001 (N \cdot m \cdot s)$
- Pole pairs: p = 3

### 4.2. Trapezoidal control method

Figure 2 shows the Simulink diagram of trapezoidal control for BLDC motors. Based on Hall-effect sensors locating a fixed point within the motor housing to monitor the rotor position, this control method has been extensively applied in practical cases. The sensor signals give feedback and allow the motor phases to be energised at angles that are synchronised and accurate. This timed switching, or commutation, is an essential element to achieving the desired direction of rotation and for effective torque production.

To adjust the motor speed, a PWM on the DC input voltage is utilised. A change in the duty ratio of the PWM will change the effective voltage applied to the motor, and the motor speed can be dynamically controlled in this way. Such a power modulation technique gives more freedom and energy efficiency to the controller in case of load variations and reference input variations. To show the control performance, the actual motor speed is compared with its commanded reference speed in Figure 3. The response indicates that the proposed controller effectively attenuates the tracking error, and the motor speed accurately tracks the reference value. The figure also plots the stator current waveform with a faster switching trapezoidal waveform.

Trapezoidal control of BLDC motors, while simple and cost-effective, has several disadvantages that limit their suitability for high-performance applications. One of the main issues is torque ripple, which leads to vibrations and audible noise due to the abrupt switching of motor phases. This results in reduced smoothness and precision, making it less ideal for applications requiring fine motion control. Additionally, efficiency drops at low speeds, especially in sensorless configurations where back-EMF detection becomes unreliable. The control method also



Figure 2. Simulink diagram of the trapezoidal control of BLDC motors. BLDC, brushless DC; EMF, electromotive force; PWM, pulse-width modulation.



Figure 3. Motor speed and stator current of the BLDC motor with the trapezoidal control method. BLDC, Brushless DC.

makes suboptimal use of the motor's magnetic field, leading to lower overall energy efficiency compared to more advanced techniques like FOC. These drawbacks make trapezoidal control less suitable for applications that demand quiet operation, high efficiency, or precise control.

### 4.3. FOC method

The Simulink model of the FOC applied to BLDC motors is illustrated in Figure 4. Unlike trapezoidal control, which requires Hall sensors for rotor position sensing, this is no longer necessary with the use of advanced mathematical transforms like Clarke and Park introduced by FOC. These transformations then decouple torque and flux components and allow for their independent and accurate control. FOC control can be seen as the most efficient current vector control and directs the current stator vector into the direction of the rotor magnetic field as a current vector, whereby a maximum torque is produced. This feature not only results in quieter, more efficient motor operation but also provides much more responsive system performance. The control system dynamically regulates the current of the motor in a real-time manner, which can adjust to the load demand requirement, together with the torque ripple, with better transient performance.

The superior dynamic response of FOC makes it especially well-suited for high-performance, high-precision applications. These applications include areas such as robotics, which depend on precise positioning and smooth motion; industrial automation systems, whose success is tied to reliability and precision of control; and, in the world of electric and hybrid vehicles, where high efficiency and responsive torque control result in improved mileage as well as improved performance. The speed response of the FOC-based system is evident in Figure 5, where it closely follows the reference speed command. As can be seen in the graph, the real speed of the motor approximates the desired speed. This highly precise tracking result ensures the robustness and accuracy of the FOC algorithm for achieving the desired operating conditions when the load or speed commands change. The small error between these reference speed and actual speed profile graphs shows the high fidelity of the system in real time as well.

In conclusion, the choice between trapezoidal control and FOC for BLDC motors depends on the specific application requirements. Trapezoidal control offers a simpler and more cost-effective solution suitable for low-performance or budget-conscious systems, despite its limitations in torque smoothness and efficiency. On the other hand, FOC provides superior performance, efficiency and smooth operation across a wide speed range, making it the preferred option for demanding and high-performance applications, albeit at the cost of increased complexity and hardware demands.



Figure 4. Simulink diagram of the FOC method of BLDC motors. BLDC, Brushless DC; FOC, field-oriented control; SVPWM, space vector PWM.

# 5. Experiment

This section provides a detailed procedure for deploying a digital PID controller for a commercial sensorless BLDC motor drive. The gains of the PID controller,  $K_p$ ,  $K_i$  and  $K_d$ , are determined using the Nelder–Mead optimisation method. The transfer function of the BLDC motor drive was first obtained to get appropriate initial values for the controller gains.

### 5.1. Hardware setup

Figure 6 presents the detailed hardware setup designed for precise motor control applications. At the core of the system is a 12V brushless BLDC motor from Nidec (Japan), which functions as the main actuator, delivering accurate rotational motion. The motor is powered by a reliable 12 V power supply that ensures consistent performance. To facilitate real-time speed measurement and enable feedback-based control, a rotary encoder is coupled with the motor shaft. Instead of using Hall effect sensors, the system employs a commercial sensorless BLDC motor driver that relies on a three-phase inverter configuration to estimate the rotor's position electronically. This reduces hardware complexity and enhances durability. The overall control and coordination of the system are handled by an Arduino Nano, a compact microcontroller built around the ATmega328P. The Arduino Nano offers a balanced of processing power, memory and input/output interfaces, making it ideal for embedded applications where space is limited but precise control is essential.



Figure 5. Motor speed and stator current of the BLDC motor with the FOC method. BLDC, brushless DC; FOC, field-oriented control.



Figure 6. The hardware. BLDC, brushless DC.

### 5.2. System identification

To identify the transfer function of a system comprising both the motor and its driver, a procedure based on the step response of the motor speed is employed. First, a 5 V step voltage is applied to the driver, which powers the motor and acts as an excitation signal to reveal the system's dynamic behaviour. Next, the motor's speed response is measured over time using an encoder, with data sampled and stored at a sufficient rate to capture the dynamics accurately. Finally, the recorded speed data is imported into MATLAB (Mathworks), where the System Identification Toolbox is used to estimate the system's transfer function. Various model structures, such as first-order or second-order transfer functions, are explored to determine the best fit for representing the combined motor and driver dynamics. By following this approach, an accurate mathematical model of the motor-driver system is obtained, which can be used for further analysis, control system design and performance optimisation. Figure 7 shows the response of the motor speed to a step voltage of 5 V applied to the control input of the BLDC motor driver.

The step input voltage and the motor speed response were used to identify the transfer function of the system, which includes both the motor and the driver, using the MATLAB System Identification Toolbox. The motor speed response was recorded with a sampling interval of 0.1 s. The transfer function of the entire system is modelled as a second-order system, as follows:

$$G_m(s) = \frac{810.8}{s^2 + 2.366s + 2.76} \tag{24}$$

### 5.3. The PID controller design

The design of an optimal PID controller for the system was conducted in a two-step process to ensure precise control, stability and optimal performance. The steps are detailed as follows:

- 1) Obtaining initial gains for the PID controller: The initial gains of the PID controller can be determined using the transfer function identified above with the following MATLAB script:
  - num = 810.8; den = [1 2.366 2.76]; Gm = tf(num,den); C1 = pidtune(Gm,'pid'); Kp1 = C1.Kp; Ki1 = C1.Ki; Kd1 = C1.Kd;



Figure 7. Response of the real motor speed to a step voltage of 5 V.

Running the MATLAB script results in the following PID controller gains:

 $K_p = 0.0073$ ,  $K_i = 0.0082$  and  $K_d = 0.0013$ 

### 2) Tuning the PID controller gains using the Nelder-Mead method:

**Defining an objective function:** The objective function is an ITAE that depends on the system transfer function and the PID controller gains and can be defined in MATLAB as follows:

```
function S = objectiveFcn(x)

s = tf('s');

Gm = 810.8/(s^2 + 2.366^*s + 2.76);

Kp = x(1);

Ki = x(2);

Kd = x(3);

Gc = pid(Kp,Ki,Kd);

T = 0.01;

t = 0:T:10;

e = 1 - step(feedback(Gc*Gm,1),t);

S = sum(t'.*abs(e));
```

3) Tuning PID controller gains using the Nelder-Mead method: The PID controller gains can be optimally obtained using the following MATLAB script:

num = 810.8; den = [1 2.366 2.76]; Gm = tf(num,den); C1 = pidtune(Gm,'pid'); Kp1 = C1.Kp; Ki1 = C1.Ki; Kd1 = C1.Kd; optimInput = [Kp1 Ki1 Kd1]; options = optimset('Display','iter','PlotFcns',@optimplotfval,'MaxIter',30); [a, fval] = fminsearch(@objectiveFcn,optimInput,options); Kp2 = a(1); Ki2 = a(2); Kd2 = a(3);

The optimal values of the PID controller gains are given by:

 $K_p = 0.0165, K_i = 0.0189$  and  $K_d = 0.0073$ 

As the sampling period, *T*, is 0.1 s, then:

a = 0.0165  $b = K_i T = 0.0189 \times 0.1 = 0.0019$  $c = \frac{K_d}{T} = \frac{0.0073}{0.1} = 0.073$ 

### 5.4. Deployment of the digital PID controller on a microcontroller

The digital PID controller can be conveniently deployed on the Arduino Nano board by using a C script as follows:

```
volatile long temp, counter;
int Ts = 100;
volatile long encoderValue = 0;
long previousMillis = 0;
long currentMillis = 0;
void setup(){
  Serial.begin (9600);
  pinMode(2, INPUT_PULLUP);
  pinMode(3, INPUT_PULLUP);
  attachInterrupt(0, ai0, RISING);
  attachInterrupt(1, ai1, RISING);
}
void ai0(){
  if (digitalRead(3) == LOW) counter++;
  else counter --;
}
void ai1(){
  if (digitalRead(2) == LOW) counter--;
  else counter++;
}
float a = 0.0165;
float b = 0.0019;
float c = 0.073;
float rk = 800;
float yk = 0;
float uk = 0;
float ek = 0;
float ek 1 = 0;
float pk = 0;
float pk_1 = 0;
float qk = 0;
void loop() {
 currentMillis = millis();
  if (currentMillis - previousMillis >= Ts){
     previousMillis = currentMillis;
    yk = (float)counter*3/4;
    Serial.println(yk);
    ek = rk - yk;
    pk = pk_1 + b * ek;
    qk = c * (ek - ek_1);
    uk = a * ek + pk + qk;
```

Nguyen et al.

```
if (uk >= 255){
    uk = 255;
    }
if (uk <= 0){
    uk = 0;
    }
    ek_1 = ek;
    pk_1 = pk;
    analogWrite(5, uk);
    counter = 0;
    }
}</pre>
```

### 5.5. Experimental results

Figure 8a–c illustrate the motor speeds corresponding to reference speeds of 500 rpm, 700 rpm and 900 rpm, respectively. These graphs highlight the system's dynamic behaviour and provide a comparative analysis of two PID control strategies. The first strategy utilises the initial PID controller gains, while the second applies the Nelder–Mead optimised PID controller gains. Based on the comparisons between the initial and optimised PID controllers, as shown in Tables 1–3, it is evident that the Nelder–Mead optimised PID controller achieves a shorter settling time, which refers to the time required for the motor speed to stabilise and reach the reference speeds. This marks a significant improvement over the initial PID controller, which takes longer to reach the desired speeds. The reduced settling time demonstrates the optimised controller's superior performance, allowing the system to respond more quickly and efficiently to changes in input or reference speed. This enhancement in response time is critical for applications requiring rapid and precise motor control, further validating the effectiveness of the Nelder–Mead optimisation method in tuning PID controller gains for improved system performance.

# 6. Conclusions

This study presents a comprehensive and methodical framework for the design of an optimised digital PID controller specifically tailored for commercial sensorless BLDC motor drives. The proposed optimisation approach leverages the Nelder–Mead simplex algorithm, a derivative-free technique well-suited for non-linear, multidimensional optimisation problems where the objective function may be discontinuous, noisy or lacking an easily computable gradient. Unlike traditional gradient-based methods, which rely on explicit derivative calculations and may struggle with non-convex or irregular search spaces, the Nelder–Mead algorithm uses a simplex of points that iteratively adapts through operations such as reflection, expansion, contraction and shrinkage. This enables efficient exploration



Figure 8. Motor speed responses for reference speeds of 500 rpm (a), 700 rpm (b) and 900 rpm (c).

Table 1. Performance comparison between the initial and optimised PID controllers at a speed of 500 rpm.

Controller	Rise time (s)	Settling time (s)	Overshoot (%)
Initial PID controller	3.5867	6.8628	0.0899
Optimised PID controller	1.6477	2.8985	0.6736

Table 2. Performance comparison between the initial and optimised PID controllers at a speed of 700 rpm.

Controller	Rise time (s)	Settling time (s)	Overshoot (%)
Initial PID controller	3.7983	6.5593	0.4619
Optimised PID controller	1.6200	3.8898	0.2130

Table 3. Performance comparison between the initial and optimised PID controllers at a speed of 900 rpm.

Controller	Rise time (s)	Settling time (s)	Overshoot (%)
Initial PID controller	3.5867	6.3841	0.1332
Optimised PID controller	1.3312	2.4900	0.2745

of the parameter space, resulting in fine-tuned PID gains that enhance dynamic response characteristics such as reduced overshoot and faster settling time. One of the core strengths of the Nelder–Mead method lies in its ease of implementation and adaptability to a wide range of real-world engineering problems, making it particularly attractive for embedded systems and digital control applications where computational resources may be limited. While this study validates the effectiveness of Nelder–Mead optimisation in tuning PID controllers for BLDC motor drives, it also lays the groundwork for future investigations aimed at comparative performance analysis. Subsequent research will focus on evaluating the relative strengths and limitations of the Nelder–Mead algorithm against other popular direct search-based optimisation techniques, such as GA and PSO. These algorithms are well-established in control system optimisation and have demonstrated significant potential in various motor control applications due to their global search capabilities and adaptability to complex optimisation landscapes.

### Acknowledgements

This research is funded by Hanoi University of Science and Technology (HUST) under project number T2024-PC-059.

### References

- Aishwarya,V. and Jayanand, B. (2016). Estimation and control of sensorless brushless DC motor drive using extended Kalman filter. In: 2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT), Nagercoil, India, 2016.
- Alberto, A. P., Michael, F. and Chunjiang, Q. (2009). Particle swarm optimization for PID tuning of a BLDC motor. In: 2009 IEEE International Conference on Systems. Man and Cybernetics, San Antonio, TX, USA, 11–14 October 2009, pp. 3917–3922.
- Basilio, J. C. and Matos, S. R. (2002). Design of PI and PID Controllers with Transient Performance Specification. *IEEE Transactions on Education*, 45(4), pp. 364–370. doi: 10.1109/TE.2002.804399
- Bazanella, A. S., Pereira, L. F. A. and Parraga, A. (2017). A New Method for PID Tuning Including Plants without Ultimate Frequency. *IEEE Transactions on Control Systems Technology*, 25(2), pp. 637–644. doi: 10.1109/TCST.2016.2557723
- Bosso, A., Conficoni, C., Raggini, D. and Tilli, A. (2021). A Computational-Effective Field-Oriented Control Strategy for Accurate and Efficient Electric

Propulsion of Unmanned Aerial Vehicles. *IEEE/ ASME Transactions on Mechatronics*, 26(3), pp.1501–1511.doi:10.1109/TMECH.2020.3022379

- Chau, K. T., Zhang, D., Jiang, J. Z., Liu, C. and Zhang, Y. (2007). Design of a Magnetic-Geared Outer-Rotor Permanent-Magnet Brushless Motor for Electric Vehicles. *IEEE Transactions on Magnetics*, 43(6), pp. 2504–2506. doi: 10.1109/TMAG.2007.893714
- Chen, G, Y. and Perng. J. W. (2017). PI speed controller design based on GA with time delay for BLDC motor using DSP. In: 2017 IEEE International Conference on Mechatronics and Automation (ICMA), Takamatsu, Japan, 6–9 August 2017, pp. 1174–1179.
- Chen, Z., Tomita, M., Doki, S. and Okuma, S. (2000). New Adaptive Sliding Observers for Position- and Velocity-Sensorless Controls of Brushless DC Motors. *IEEE Transactions on Industrial Electronics*, 47(3), pp. 582–591. doi: 10.1109/41.847899
- Damodharan, P. and Vasudevan, K. (2010). Sensorless Brushless DC Motor Drive Based on the Zero-Crossing Detection of Back Electromotive Force (EMF) from the Line Voltage Difference. *IEEE Transactions on Energy Conversion*, 25(3), pp. 661–668. doi: 10.1109/TEC.2010.2041781
- De, A., Stewart-Height, A. and Koditschek, D. E. (2019). Task-Based Control and Design of a BLDC Actuator for Robotics. *IEEE Robotics and Automation Letters*, 4(3), pp.2393–2400. doi: 10.1109/LRA.2019.2894860
- Gabriel, H. and Nesimi, E. (2016). Wide Speed Range Sensorless Operation of Brushless Permanent-Magnet Motor Using Flux Linkage Increment. *IEEE Transactions on Industrial Electronics*, 63(7), pp. 4052–4060. doi: 10.1109/TIE.2016.2544250
- Gopan, V. K. and Shree, J. D. (2022). Implementation of a High Power Quality BLDC Motor Drive Using Bridgeless DC to DC Converter with Fuzzy Logic Controller. *Engineering, Technology & Applied Science Research*, 12(5), pp. 9178–9185. doi: 10.48084/etasr.5213
- Gujjar, M, N. and Kumar, P. (2017). Comparative analysis of field oriented control of BLDC motor using SPWM and SVPWM techniques. In: 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), Bangalore, India, 2017, pp. 924–929.
- Ho, T, Y., Chen, M, S., Lin, J, S., Chen, P, H. (2012). The design and implementation of the BLDC motor drive for a washing machine. In: *The 1st IEEE Global Conference on Consumer Electronics*, Tokyo, Japan, 2012, pp. 156–157.

- Huang, C. L., Lee, F. C., Liu, C. J., Chen, J. Y., Lin, Y. J. and Yang, S. C. (2022). Torque Ripple Reduction for BLDC Permanent Magnet Motor Drive Using DC-Link Voltage and Current Modulation. *IEEE Access*, 10, pp. 51272–51284. doi: 10.1109/ACCESS.2022.3173325
- Huba, M., Bisták, P., Briežnik, J. and Vrancic, D. (2024). Constrained Series PI, PID and PIDA Controller Design Inspired by Ziegler–Nichols. *Power Electronics and Drives*, 9(44), pp. 331–346. doi: 10.2478/pead-2024-0021
- Khan, K. R. and Miah, M. S. (2020). Fault-Tolerant BLDC Motor-Driven Pump for Fluids with Unknown Specific Gravity: An Experimental Approach. *IEEE Access*, 8, pp. 30160–30173. doi: 10.1109/ ACCESS.2020.2972942
- Lee, T. Y., Seo, Y. K., Kim, Y. J. and Jung, S. Y. (2016). Motor Design and Characteristics Comparison of Outer-Rotor-Type BLDC Motor and BLAC Motor Based on Numerical Analysis. *IEEE Transactions on Applied Superconductivity*, 26(4), pp. 1–6. doi: 10.1109/TASC.2016.2548079.
- Mohanraj, D., Aruldavid, R., Verma, R., Sathiyasekar, K., Barnawi, A. B. and Chokkalingam, B. (2022). A Review of BLDC Motor: State of Art, Advanced Control Techniques, and Applications. *IEEE Access*, 10, pp. 54833–54869. doi: 10.1109/ ACCESS.2022.3175011
- Nelder, J. A. and Mead, R. (1965). A Simplex Method for Function Minimization. *The Computer Journal*, 7, pp. 308–313. doi: 10.1093/comjnl/7.4.308
- Owusu, G., Annan, J. K. and Nunoo, S. (2023). Neural Network-Based Optimisation of Sinusoidal PWM Controller for VSI-Driven BLDC Motor. *Power Electronics and Drives*, 8, pp. 275–298. doi: 10.2478/ pead-2023-0018
- Ozturk, S. B. and Toliyat, H. A. (2011). Direct Torque and Indirect Flux Control of Brushless DC Motor. *IEEE/ASME Transactions on Mechatronics*, 16(2), pp. 351–360. doi: 10.1109/TMECH.2010.2043742
- Pakdeeto, J., Wansungnoen, S., Areerak, K. and Areerak, K.
  (2023). Optimal Speed Controller Design of Commercial BLDC Motor by Adaptive Tabu Search Algorithm. *IEEE Access*, 11, pp. 79710–79720. doi: 10.1109/ACCESS.2023.3300233
- Raja, S. and Rathinakumar, M. (2023). Transient Analysis of the Fuzzy Logic-based Speed Control of a Three-phase BLDC Motor. *Engineering, Technology & Applied Science Research*, 13(1), pp. 9855–9860. doi: 10.48084/etasr.5419
- Rubaai, A. and Young, P. (2015). Hardware/Software Implementation of Fuzzy-Neural-Network Self-Learning Control Methods for Brushless DC

Motor Drives. *IEEE Transactions on Industry Applications*, 52(1), pp. 414–424. doi: 10.1109/ TIA.2015.2468191

- Shao, J. (2006). An Improved Microcontroller-Based Sensorless Brushless DC (BLDC) Motor Drive for Automotive Applications. *IEEE Transactions* on Industry Applications, 42(5), pp. 1216–1221. doi: 10.1109/TIA.2006.880888
- Sinlapakun, V. and Assawinchaichote, W. (2015). Optimized PID Controller Design for Electric

Furnace Temperature Systems With Nelder Mead Algorithm. In: 2015 12th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), Hua Hin, Thailand, 2015.

Ubare, P. and Sonawane, D. N. (2022). Performance Assessment of the BLDC Motor in EV Drives using Nonlinear Model Predictive Control. *Engineering, Technology & Applied Science Research*, 12(4), pp. 8901–8909. doi: 10.48084/etasr.4976