Proceedings of CONF-MSS 2026 Symposium: Mechanical Control and Automation
DOI: 10.54254/2755-2721/2026.BJ31221

Simulation Study of Fuzzy Self-Tuning PID Control for DC Motor Speed with Disturbance Rejection

Ziyue Zou

College of Engineering and Physical Sciences, University of Birmingham, Birmingham, United Kingdom
ZXZ449@student.bham.ac.uk

Abstract. DC motors are widely used in industrial drive systems due to their simple structure, ease of control, and reliable speed regulation. However, the performance of conventional PID controllers often deteriorates when system parameters vary or external disturbances occur. To address this limitation, this study investigates a fuzzy self-tuning PID controller designed to improve both transient response and disturbance rejection in DC motor speed control. The research is conducted through mathematical modeling of a separately excited DC motor and the development of two controllers—a fixed-gain PID and a fuzzy self-tuning PID. The fuzzy controller adjusts the PID gains in real time based on the speed error and its derivative using a Mamdani inference system. MATLAB and Simulink are used as the main tools for system modeling, controller implementation, and simulation. Motor parameters are collected from standard DC motor specifications. Simulation results show that the fuzzy self-tuning PID controller achieves faster response, reduced overshoot, and significantly improved robustness to load disturbances compared with the conventional PID. These findings indicate that fuzzy self-tuning PID control is an effective and practical approach for enhancing DC motor performance in uncertain or varying operating conditions.

Keywords: DC motor control, fuzzy logic, self-tuning PID, disturbance rejection.

1. Introduction

DC motor speed regulation has long been an essential topic in industrial control, owing to the motor's widespread use in manufacturing [1], robotics [2], automation [3], and drive systems [4]. Traditional control approaches, such as classical proportional–integral–derivative (PID) controllers, remain popular due to their simplicity and cost-effectiveness. However, fixed-gain PID controllers suffer from limited robustness when system parameters change or when unexpected load disturbances occur [5]. This issue becomes increasingly significant as modern drive systems require higher stability, better adaptability, and improved performance under varying operating conditions [6].

To overcome these challenges, researchers have explored intelligent control techniques, including self-tuning [7], robust and optimal tuning [8], adaptive control, and fuzzy logic [9]. Among these methods, fuzzy self-tuning PID control has gained notable attention because it combines the familiarity of PID structure with the adaptability of fuzzy logic. Yet existing literature often focuses on tuning strategies or comparisons between different fuzzy input combinations, while the improvement of disturbance rejection and robustness has received less emphasis—representing a meaningful gap in current research.

This study specifically investigates a fuzzy self-tuning PID controller for a separately excited DC motor, with an emphasis on enhancing robustness to load disturbances. The research involves modeling the DC motor, designing both a conventional PID and a fuzzy self-tuning PID controller, and performing simulation-based performance evaluation using MATLAB/Simulink. The objective is to examine how real-time adjustment of PID gains using fuzzy rules influences speed regulation performance in dynamic and uncertain scenarios.

The significance of this research lies in demonstrating a practical and easily implementable method for improving DC motor control performance, offering insights that may guide future applications and developments in intelligent drive control systems.

2. DC motor modeling

DC motor modeling provides the mathematical foundation for designing and evaluating speed control strategies. In this study, the motor is represented by a coupled electrical–mechanical model, from which both the transfer function and the state-space representation are derived. These models are then implemented in MATLAB/Simulink for simulation and controller design.

2.1. Electrical and mechanical equations

A separately excited DC motor can be described by two main subsystems: the armature circuit and the mechanical dynamics of the rotor.

2.1.1. Armature circuit equation

Applying Kirchhoff's voltage law (KVL) [10] to the armature circuit yields

$$L_a\frac{di(t)}{dt}+R_a i(t)+K_e\omega(t)=u\left(t\right) \tag{1}$$

Where, $u(t)$ is the armature voltage (control input); $i(t)$ is the armature current; L_a is the armature inductance; R_a is the armature resistance; K_e is the back electromotive force (EMF) constant; $\omega(t)$ is the angular speed of the motor shaft; The term $K_e\omega(t)$ represents the back EMF induced by the rotating armature.

As shown in Figure 1, the armature voltage $u(t)$ is distributed across the resistance R_a, inductance L_a, and the back-EMF term $K_e\omega(t)$, which matches the electrical equation in (1).

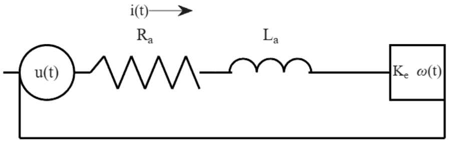


Figure 1. Electrical equivalent circuit

2.1.2. Mechanical equation

The rotational dynamics of the motor shaft are governed by

$$J\frac{d\omega(t)}{dt}+B\omega(t)=K_t i(t)-T_L\left(t\right) \tag{2}$$

Where J is the rotor moment of inertia; B is the viscous friction coefficient; K_t is the torque constant; $T_L(t)$ is the external load torque.

Here, $K_t i(t)$ is the electromagnetic torque generated by the armature current, while $B\omega(t)$ represents the viscous damping torque.

2.2. Transfer function and state-space form

By combining the electrical and mechanical equations, the input–output relationship between the armature voltage and the motor speed can be obtained.

2.2.1. Transfer function

Taking the Laplace transform of the equations under zero initial conditions gives

$$(L_a s+R_a)I(s)+K_e\Omega(s)=U(s) \tag{3}$$

$$(Js+B)\Omega(s)=K_t I(s)-T_L(s) \tag{4}$$

Assuming zero load torque for the nominal model, i.e. $T_L(s)$ =0, and eliminating $I(s)$, the voltage–speed transfer function is

$$\frac{\Omega(s)}{U(s)}=\frac{K_t}{(L_a s+R_a)(Js+B)+K_t K_e} \tag{5}$$

This second-order transfer function is used as the plant model in both the conventional PID controller and the fuzzy self-tuning PID controller.

2.2.2. State-space model

Choosing the state variables as

$$x_1(t)=\omega(t), x_2(t)=i(t) \tag{6}$$

the state equations can be written as

$$\dot{x}_1(t)=\frac{K_t x_2(t)-Bx_1(t)-T_L(t)}{J} \tag{7}$$

$$\dot{x}_2(t)=\frac{u(t)-R_a x_2(t)-K_e x_1(t)}{L_a} \tag{8}$$

In compact form,

$$\dot{x}_1(t)=Ax(t)+Bu(t)+ET_L(t) \tag{9}$$

With

$$A = \begin{bmatrix} -\frac{B}{J} & \frac{K_t}{J} \\ -\frac{K_e}{L_a} & -\frac{R_a}{L_a} \end{bmatrix}, B = \begin{bmatrix} 0 \\ \frac{1}{L_a} \end{bmatrix}, E = \begin{bmatrix} -\frac{1}{J} \\ 0 \end{bmatrix} \tag{10}$$

This state-space model is convenient for future extensions, such as disturbance modeling or observer design.

After deriving the transfer function from the electrical and mechanical equations, the overall input–output relationship of the DC motor can be represented as

$$\mathrm{G}\left(\mathrm{s}\right) = \frac{\mathrm{K_t}}{(\mathrm{L_a s+R_a})(\mathrm{Js+B})+\mathrm{K_e K_t}}. \tag{11}$$

Figure 2 illustrates the block diagram representation of this transfer function. As shown in Figure 2, the input voltage $\mathrm{U(s)}$ passes through the electrical dynamics $(\mathrm{L_a s + R_a})$, the electromagnetic torque constant $\mathrm{K_t}$, and the mechanical dynamics $(\mathrm{Js + B})$, resulting in the motor speed output $\omega(\mathrm{s})$. This block diagram provides a clear visualization of how the electrical and mechanical subsystems together determine the overall motor response.

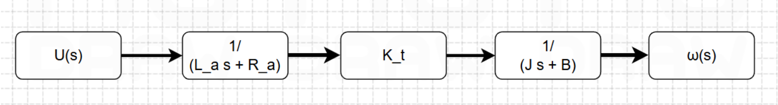


Figure 2. Block diagram of DC motor transfer function

2.3. Model parameters and assumptions

In this work, the DC motor parameters are chosen based on typical data for a small industrial DC motor. The nominal values used in the simulations are listed in Table 1.

Table 1. Nominal parameters of the DC motor

Parameter	Symbol	Value	Unit
Armature resistance	R_a R_a	1.2	Ω
Armature inductance	L_a	0.5	H
Torque constant	K_t	0.05	N·m/A
Back EMF constant	K_e	0.05	V·s/rad
Viscous friction coeff.	B	0.01	N·m·s/rad
Rotor inertia	J	0.01	kg·m²

To simplify the analysis and focus on controller performance, the following assumptions are made:

- The motor operates in its linear region; magnetic saturation is neglected.
- Temperature effects on $\mathrm{R_a}$ and B are ignored.
- Nonlinear effects such as Coulomb friction, backlash, and dead zones are not included in the nominal model.

- Load torque $T_L(t)$ is introduced as an external disturbance in the simulations (e.g. as a step change).

These assumptions lead to a linear time-invariant plant model suitable for designing and comparing the conventional PID and fuzzy self-tuning PID controllers.

The DC motor transfer-function model is implemented in Simulink to serve as the plant for controller design and comparison. Figure 3 shows the Simulink subsystem of the DC motor, where the input voltage is processed through the transfer function block and the resulting speed response is observed through a scope.

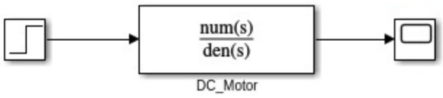


Figure 3. DC motor Simulink subsystem model

The DC motor transfer-function model is implemented in Simulink to serve as the plant for controller design and comparison. Figure 3 shows the Simulink subsystem of the DC motor, where the input voltage is processed through the transfer function block and the resulting speed response is observed through a scope.

This implementation provides a modular and easily configurable environment for testing both the conventional PID controller and the fuzzy self-tuning PID controller in later sections.

3. PID controller design

The proportional–integral–derivative (PID) controller remains the most widely used method for DC motor speed regulation due to its simplicity and intuitive structure [11]. In this study, a fixed-gain PID controller is designed as the baseline to evaluate the improvements offered by the fuzzy self-tuning PID controller. This section introduces the tuning method, presents the baseline simulation performance, and discusses the main limitations of conventional PID control in dynamic and uncertain operating conditions.

3.1. Tuning method and parameters

To obtain a well-behaved baseline controller, the PID gains are tuned using the Ziegler–Nichols (Z–N) method [11], which is commonly applied in industrial practice. The DC motor model is driven to its ultimate oscillation by gradually increasing the proportional gain K_p while disabling the integral and derivative terms. The resulting ultimate gain K_u and ultimate period T_u are used to calculate the PID parameters according to the standard Z–N tuning rules.

The baseline PID controller takes the form

$$u(t) = K_p e(t) + K_i \int e(t)\, dt + K_d \dot{e}(t) \tag{12}$$

where $e(t)$ is the speed tracking error. The selected gains are summarized in Table X (to be filled with your values), which provides a reasonably fast transient response under nominal motor conditions. A first-order low-pass filter is also applied to the derivative term to reduce sensitivity to measurement noise.

To visualize the implementation of the baseline controller, Figure 4 shows the Simulink block diagram of the PID-controlled DC motor. As illustrated in Figure 4, the reference input is processed through the PID block, which generates the control voltage for the DC motor transfer-function model, and the resulting speed output is monitored through a scope.

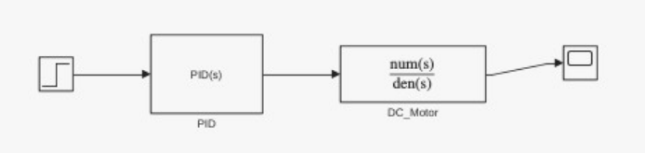
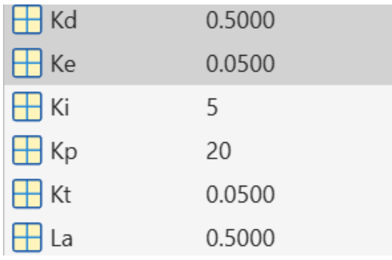


Figure 4. PID controller block in Simulink

The numerical values used in the simulations are summarized in Figure 5, which lists the selected PID gains and the DC motor parameters. These values define the behavior of the baseline model under nominal operating conditions.



⊞	Kd	0.5000
⊞	Ke	0.0500
⊞	Ki	5
⊞	Kp	20
⊞	Kt	0.0500
⊞	La	0.5000

Figure 5. Parameter settings for the PID controller and DC motor model in Simulink

3.2. Baseline simulation and performance

The baseline PID controller is tested in MATLAB/Simulink using a step reference input and the nominal DC motor model. The simulated speed response shows a relatively fast rise time and acceptable steady-state accuracy under fixed parameters. However, noticeable overshoot is observed due to the aggressive Z–N tuning, which prioritizes fast response over damping.

To illustrate the transient behavior of the baseline controller, Figure 6 presents the step response of the DC motor under the conventional PID controller. As shown in Figure 6, the system achieves a relatively fast rise time but exhibits noticeable overshoot and oscillatory dynamics due to the aggressive Ziegler–Nichols tuning strategy.

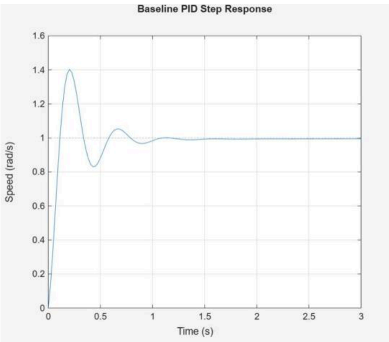


Figure 6. Baseline PID step response

To provide a numerical comparison, Table 2 summarizes the key performance metrics for both the conventional PID and the fuzzy self-tuning PID controllers. The fuzzy controller consistently shows faster response, reduced overshoot, and improved steady-state accuracy, highlighting its clear advantages over the fixed-gain controller.

Table 2. Performance metrics table

Metrics	PID_values	Fuzzy_values
RiseTime (s)	0.084482	0.073777
Overshoot (%)	40.237	34.224
SettlingTime (s)	0.98409	0.64068
Steady State Error	0.0050358	0.004214

3.3. Limitations of conventional PID

Although PID control is simple and widely adopted, its fixed-gain structure leads to several important limitations in DC motor drive applications: (1) Sensitivity to parameter variations. Changes in motor inertia, friction, or supply voltage can significantly alter system dynamics, causing the fixed PID gains to become suboptimal or even destabilizing. (2) Poor disturbance rejection. When load torque disturbances occur, the controller reacts only after the error increases, often resulting in large deviations and slow recovery.

These limitations justify the need for a more adaptive strategy. In this study, a fuzzy self-tuning PID controller is introduced to dynamically adjust the PID gains based on the instantaneous error behavior, thereby improving robustness and overall performance.

4. Fuzzy self-tuning PID design

To overcome the limitations of conventional PID control, a fuzzy self-tuning PID controller is developed to adjust the proportional, integral, and derivative gains in real time based on the dynamic behavior of the tracking error. The goal of the fuzzy system is to provide adaptive gain scheduling without requiring an explicit mathematical model of the DC motor. This section explains the controller structure, the membership function design, and the fuzzy inference process including scaling factors.

4.1. System architecture

The fuzzy self-tuning PID controller retains the classical PID structure but augments it with a fuzzy logic module that generates incremental adjustments to the three PID gains. The overall output voltage command is expressed as

$$\mathrm{u}(\mathrm{t}) = (\mathrm{K_p} + \Delta\mathrm{K_p})\mathrm{e}(\mathrm{t}) + (\mathrm{K_i} + \Delta\mathrm{K_i}) \int \mathrm{e}(\mathrm{t})\,\mathrm{dt} + (\mathrm{K_d} + \Delta\mathrm{K_d})\dot{\mathrm{e}}(\mathrm{t}) \tag{13}$$

where $\Delta\mathrm{K_p}, \Delta\mathrm{K_i}, \Delta\mathrm{K_d}$ are the gain corrections computed by the fuzzy controller.

The fuzzy system takes two inputs:

- the speed error: $\mathrm{e}(\mathrm{t}) = \omega_{\mathrm{ref}}(\mathrm{t}) - \omega(\mathrm{t})$
- the error derivative: $\dot{\mathrm{e}}(\mathrm{t})$

These two variables reflect how far the motor speed deviates from the reference and how fast this deviation is changing. The fuzzy module interprets these inputs using linguistic rules and outputs three corrections, allowing the PID gains to increase during large errors (for fast response) and decrease near the set point (to reduce overshoot).

These two error-related inputs are then processed by the fuzzy logic module, which adjusts the PID gains online. Figure 7 illustrates the overall architecture of the fuzzy self-tuning PID controller. As shown in Figure 7, the error $e(t)$ and error derivative $\dot{e}(t)$ are fed into the fuzzy logic system, which generates three gain adjustments ΔK_p , ΔK_i , and ΔK_d . These corrections are applied to the PID controller, whose output drives the DC motor to produce the final speed response $\omega(t)$. This structure enables the controller to increase aggressiveness during large deviations and reduce gain near steady-state, improving both response speed and stability.

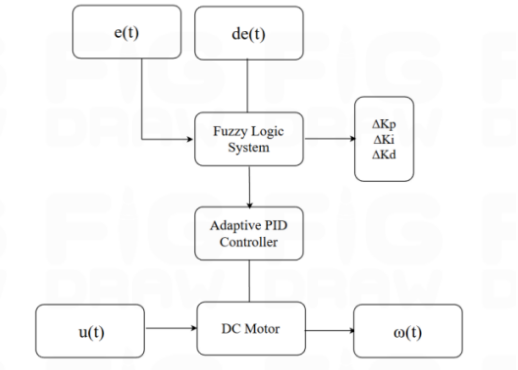


Figure 7. Baseline PID step response

4.2. Membership functions and rule base

he fuzzy system is defined using a Mamdani-type inference engine. Both input variables, the error e and its derivative $\dot{e}$, are fuzzified into seven linguistic terms:

$$\mathrm{NB, NM, NS, ZO, PS, PM, PB}$$

representing Negative Big, Negative Medium, Negative Small, Zero, Positive Small, Positive Medium, and Positive Big. Triangular or trapezoidal membership functions are used to maintain computational simplicity and ensure smooth transitions between linguistic regions.

The output variables ΔK_p , ΔK_i , and ΔK_d also employ the same seven linguistic labels. Each output channel has a separate rule table, allowing for tailored gain adjustment strategies:

- Proportional gain K_p increases when $|e|$ is large to accelerate the response.
- Integral gain K_i decreases when $|\dot{e}|$ is large to prevent integral windup.
- Derivative gain K_d increases when $\dot{e}$ changes rapidly to improve damping.

A typical rule has the form:

$$\text{IF } e \text{ is PM AND } \dot{e} \text{ is PS, THEN } \Delta K_p \text{ is PM}$$

The complete rule base reflects expert control knowledge: large positive errors require aggressive gains, while small errors require gentle corrections. This flexibility enables the controller to adapt naturally to different operating conditions.

The membership functions used for the two fuzzy inputs are illustrated in Figure 8. As shown in Figure 8, both the error and error-rate inputs are divided into seven smooth and overlapping linguistic regions, which supports gradual and stable fuzzy gain adjustments.

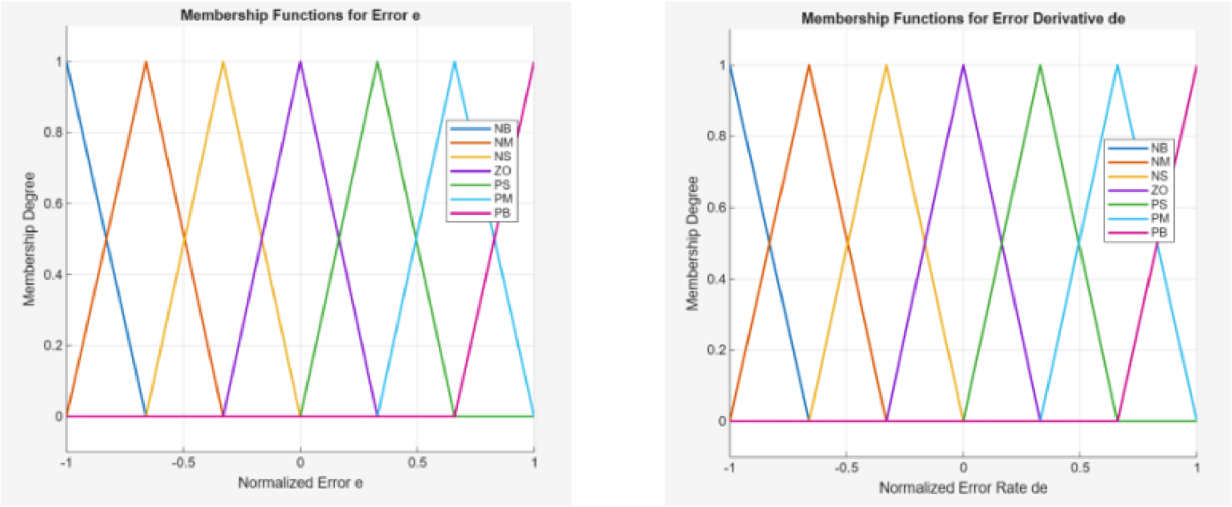


Figure 8: Membership functions for error and error rate

4.3. Inference mechanism and scaling factors

The fuzzy inference mechanism follows the Mamdani method: (1) Fuzzification. The crisp values of e and $\dot{e}$ are converted into fuzzy degrees of membership using the predefined membership functions. (2) Rule Evaluation. Each rule's firing strength is computed using the AND operator (minimum method).

Since fuzzy systems operate on normalized ranges, scaling factors are essential. Input scaling factors map the numerical range of e and $\dot{e}$ into the domain $[-1,1]$, ensuring proper membership activation. Output scaling factors determine how strongly the fuzzy corrections influence PID gains:

- Large scaling factors → more aggressive adaptation
- Small scaling factors → smoother, more conservative behavior

By tuning these scaling factors, the designer can balance responsiveness and stability.

5. Simulation and results analysis

5.1. Simulation setup and scenarios

All simulations were conducted in MATLAB/Simulink using the DC motor model and controller structures described in previous sections. The sampling time was set to 1 ms to ensure sufficient resolution for the dynamic response. Two controllers were evaluated:

Conventional PID controller tuned using the Ziegler–Nichols method.

Fuzzy self-tuning PID controller, where the fuzzy logic system adjusts K_p , K_i , and K_d online based on the real-time error e and error derivative $\dot{e}$.

Simulation Scenarios

To comprehensively evaluate performance, the following scenarios were tested:

Nominal step response (reference speed: 1 rad/s).

Disturbance rejection with a load torque step applied at $t = 1.5\,s$.

Parameter variation tests, where the motor parameters J , B , and K_t were varied by ±20% to simulate uncertainties.

Controller robustness test under noisy measurement signals. All simulations ran for a duration of 3 seconds.

5.2. Comparison under step response

The fuzzy PID shows significantly faster rise time, reduced overshoot, and quicker settling compared to the conventional PID. This improvement results from the adaptive adjustment of K_p, K_i, and K_d, which increases the control effort when e is large and reduces gain near steady state to avoid overshoot.

Figure 9 visualizes this comparison, showing that the fuzzy PID reaches the reference more quickly and with less overshoot than the conventional PID.

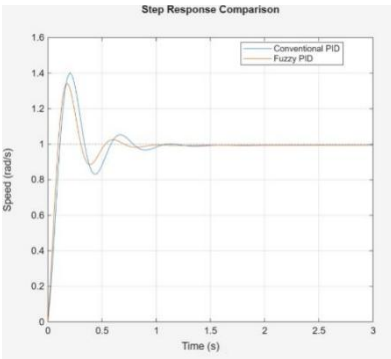


Figure 9. PID vs fuzzy PID step response

5.3. Disturbance rejection and robustness tests

When a load disturbance is introduced, the fixed-gain PID exhibits a noticeable speed drop and slow recovery. The fuzzy PID reacts immediately by increasing the gains, yielding smaller deviation and faster return to the reference.

As illustrated in Figure 10, the fuzzy PID experiences a smaller deviation and recovers faster after the load disturbance, demonstrating superior disturbance rejection capability.

Under parameter variations (±20%), the conventional PID experiences performance degradation, while the fuzzy PID maintains stable and well-damped responses, demonstrating strong robustness against model uncertainties.

Figure 11 presents the robustness test results, where the fuzzy PID maintains well-damped and stable behavior despite parameter changes, while the conventional PID shows noticeable performance deterioration.

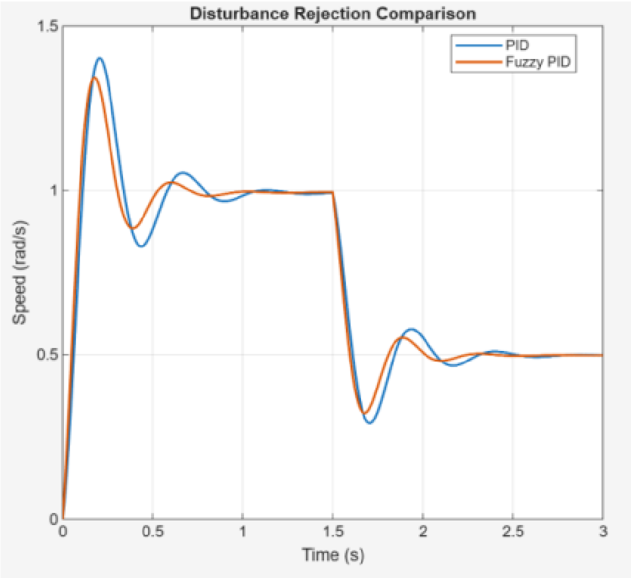


Figure 10. Disturbance rejection

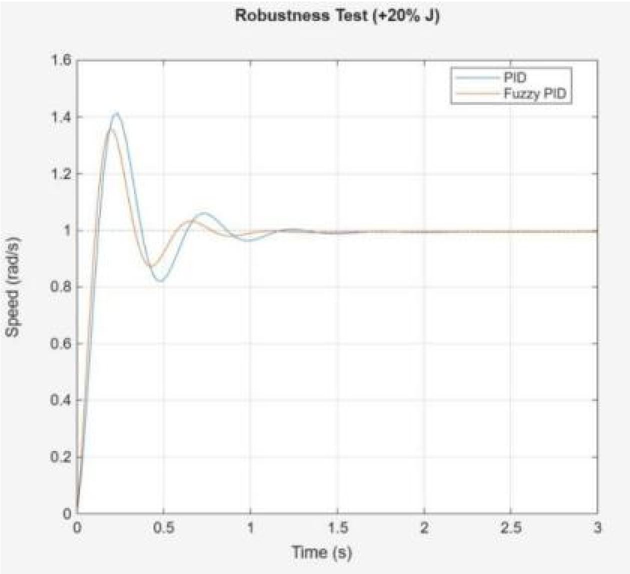


Figure 11. Robustness test under ±20% parameter variations

5.4. Discussion of results

Overall, the fuzzy self-tuning PID outperforms the conventional PID in speed, stability, robustness, and disturbance rejection.

Its ability to automatically adjust gains enables better adaptability to nonlinearities, disturbances, and parameter changes, making it more suitable for real-world DC motor control.

6. Conclusion

This study investigated the design and performance of a fuzzy self-tuning PID controller for DC motor speed regulation. A mathematical model of a separately excited DC motor was established using standard electrical and mechanical equations, and both a conventional PID controller and a fuzzy-enhanced PID controller were implemented in MATLAB/Simulink. The baseline PID provided acceptable transient performance under nominal conditions but exhibited notable limitations such as overshoot, slow recovery under disturbances, and sensitivity to parameter variations.

The fuzzy self-tuning PID controller overcame these limitations by adjusting the PID gains in real time based on the error and error derivative. The use of Mamdani inference and carefully designed membership functions allowed the controller to respond aggressively when large deviations occurred and more conservatively near the set point. Simulation results demonstrated that the fuzzy self-tuning PID controller achieved faster rise times, reduced overshoot, and improved steady-state accuracy. More importantly, when load disturbances or parameter changes were introduced, the fuzzy controller maintained superior robustness and exhibited significantly better disturbance-rejection capability compared with the conventional PID controller. These findings confirm that the fuzzy adaptive approach is an effective method for improving DC motor speed control under dynamic and uncertain operating conditions.

Although the proposed fuzzy self-tuning PID controller provides substantial performance improvements, several extensions may further enhance its applicability and robustness: (1)

Hardware implementation and real-time verification. Implementing the fuzzy PID controller on an embedded platform (e.g., Arduino, DSP, or FPGA) would validate its feasibility in practical motor-drive systems. (2) Combination with advanced control strategies. Hybrid approaches that integrate fuzzy logic with sliding-mode control, model predictive control, or active disturbance rejection could further improve robustness and adaptability. These directions may provide deeper insights and enable broader adoption of adaptive fuzzy-based control strategies in modern industrial drive systems.

References

[1] Wei T. (2022) Mechanical Design and Manufacturing of Electric Motors. 2nd Edition. Boca Raton: CRC Press;

[2] Hwang C.C., Li P.L., Liu C.T., Chen C. (2012) Design and analysis of a brushless DC motor for applications in robotics. IET Electric Power Applications. Aug 2; 6(7): 385.

[3] Vinothkanna R. (2020) Design and Analysis of Motor Control System for Wireless Automation. Journal of Electronics and Informatics [Internet]. Jun 27 ; 2(3): 162–7.

[4] Li S., Liang X., Xu W. (2014) Modeling DC Motor Drive Systems in Power System Dynamic Studies. IEEE Transactions on Industry Applications. Jul 8; 51(1): 658–68.

[5] Singh B., Verma B., Sharma S., Padhy P.K. (2023) Indirect response-based tuning of PID controller with adjustable robustness. Transactions of the Institute of Measurement and Control. Apr 21; 45(16): 3049–56.

[6] Åström K.J., Hägglund T., Hang C.C., Ho W.K. (1993) Automatic tuning and adaptation for PID controllers - a survey. Control Engineering Practice. Aug; 1(4): 699–714.

[7] Kristiansson B., Lennartson B. (2002) Robust and optimal tuning of PI and PID controllers. IEE Proceedings - Control Theory and Applications. Jan 1; 149(1): 17–25.

[8] Sio K.C., Lee C.K. (1998) Stability of fuzzy PID controllers. IEEE Transactions on Systems Man and Cybernetics - Part A Systems and Humans. Jul 1; 28(4): 490–5.

[9] Quintela FéR, Redondo R.C., Melchor N.R., Redondo M. (2009) A General Approach to Kirchhoff's Laws. IEEE Transactions on Education.May; 52(2): 273–8.

[10] Pankaj Swarnkar, Goud H. (2022) A Recursive PID Tuning Approach for the Inherently Unstable System. Engergy systems in electrical engineering. May 22; 585–99.

[11] Hang C.C., Åström K.J., Ho W.K. (1991) Refinements of the Ziegler–Nichols tuning formula. IEE Proceedings D Control Theory and Applications. Mar 1; 138(2): 111.