African Journal of Advanced Pure and Applied Sciences (AJAPAS)
Online ISSN: 2957-644X
Volume 5, Issue 1, 2026
Page No: 394-399
Website: https://aaasjournals.com/index.php/ajapas/index

ISI 2025: 1.126	SJIFactor 2024: 6.752	معامل التأثير العربي: 1.62

Design and Simulation of an Intelligent PID Controller Using Neural Network Tuning and Industrial Disturbance Modeling in MATLAB

Salim Abdallah Alhussain [1*], Mahfoud Y. A. Abdulrazigh [2]
[1,2] Department of Electrical and Electronic Engineering, College of Technical Sciences, Derna, Libya

تصميم ومحاكاة متحكم *PID* ذكي قائم على ضبط الشبكات العصبية ونمذجة الاضطرابات الصناعية باستخدام *MATLAB*

سليم عبد الله سليم الحسين [1*] ، محفوظ يونس اشتيوي [2]
[1،2] قسم الهندسة الكهربائية، كليه العلوم التقنية، درنه، ليبيا

[*]Corresponding author: saleemelhussain@gmail.com

Received: December 13, 2025	Accepted: February 26, 2026	Published: March 09, 2026

Abstract

This paper presents the design and simulation of an intelligent PID controller integrated with neural network-based tuning and industrial disturbance modeling using MATLAB. A graphical user interface (GUI) was developed to allow interactive adjustment of PID parameters (Kp, Ki, Kd) and automatic tuning via a trained artificial neural network (ANN). The system targets a first-order industrial process model, such as a liquid tank, and introduces simulated disturbances to evaluate controller robustness. The ANN was trained on performance data from various PID configurations to predict optimal gains that minimize steady-state error and improve dynamic response. The simulation includes both reference tracking and disturbance rejection scenarios, offering a realistic industrial control environment. Results demonstrate that the AI-enhanced PID controller outperforms manual tuning in terms of settling time, overshoot reduction, and adaptability to external changes. This hybrid approach bridges classical control theory with modern AI techniques, providing a flexible and intelligent solution for industrial automation and educational applications.

Keywords: PID Control- Neural Network Tuning- Artificial Intelligence- MATLAB Simulation - Industrial Process Control - Disturbance Rejection- Adaptive Control Systems- Feedforward Neural Network (FNN)- Intelligent Automation- GUI-Based Control Design.

الملخص

يعرض هذا البحث تصميم ومحاكاة متحكم PID ذكي مدمج مع أليه ضبط تعتمد على الشبكات العصبية الاصطناعية ونمذجة الاضطرابات العصبية باستخدام MATLAB.تم تطوير واجه رسوميه تفاعليه (GUI)تتيح للمستخدم تغيير معاملات المتحكم (Kp, Ki, Kd) يدويا، إضافة الي الضبط التلقائي عبر شبكة اصطناعه مدربه. يستهدف النظام نموذجًا صناعيًا من الدرجة الأولى مثل خزان السوائل، مع إدخال اضطرابات محاكاة لتقييم قدرة المتحكم على مقاومة التغيرات. تم تدريب الشبكة العصبية على بيانات الأداء الناتجة عن تكوينات مختلفة لمتحكم PID بهدف التنبؤ بالقيم المثلى للمعاملات التي تقلل الخطأ في الحالة المستقرة وتحسن الاستجابة الديناميكية. تتضمن المحاكاة سيناريوهات تتعلق بتتبع الإشارة المرجعية ورفض الاضطرابات، مما يوفر بيئة تحكم صناعية واقعية. أظهرت النتائج أن المتحكم المعزز بالذكاء الاصطناعي يتفوق على الضبط اليدوي من حيث زمن الاستقرار، تقليل التجاوز، والقدرة على التكيف مع التغيرات الخارجية. يجمع هذا النهج الهجين بين النظرية الكلاسيكية للتحكم وتقنيات الذكاء الاصطناعي الحديثة، مما يوفر حلاً مرنًا وذكيًا للتطبيقات الصناعية والتعليمية.

الكلمات المفتاحية: التحكم باستخدام PID, ضبط المعاملات بواسطة الشبكات العصبية، الذكاء الاصطناعي، المحاكاة باستخدام MATLAB, التحكم في العمليات الصناعية، أنظمة التحكم التكيفية، تصميم التحكم المعتمد على واجهة رسومية (GUI).

1. Introduction

Proportional–Integral–Derivative (PID) controllers are among the most widely used control strategies in industrial automation due to their simplicity, robustness, and effectiveness in regulating dynamic systems [1]. They are commonly applied in processes such as temperature regulation, liquid-level control, and motor speed adjustment [2]. Despite their widespread adoption, PID controllers require precise tuning of three parameters—proportional gain (Kp), integral gain (Ki), and derivative gain (Kd)—to achieve optimal performance under varying operating conditions [3].

Traditional tuning methods, including Ziegler–Nichols, Cohen–Coon, and manual trial-and-error, often fail to deliver satisfactory results in nonlinear, time-varying, or disturbance-prone environments [4]. These limitations have motivated researchers to explore intelligent tuning techniques that can adapt to changing system dynamics and improve control accuracy.

Artificial intelligence (AI), particularly artificial neural networks (ANNs), has emerged as a powerful tool for enhancing classical control systems. ANNs are capable of learning complex input–output relationships and generalizing control strategies based on training data [5]. When integrated with PID controllers, neural networks can automate the tuning process by predicting optimal gain values that minimize performance metrics such as overshoot, settling time, and steady-state error [6].

This paper presents the design and simulation of an intelligent PID controller using MATLAB. A feedforward neural network (FNN) is trained on a dataset of PID configurations and system responses to predict suitable gain values. A graphical user interface (GUI) was developed to allow users to manually adjust PID parameters or apply AI-based tuning with a single click. The system simulates a first-order industrial process, such as a liquid tank, and introduces external disturbances to evaluate the controller's robustness. The proposed approach demonstrates how combining classical control theory with AI techniques can yield a flexible, adaptive, and intelligent control solution for industrial automation and educational environments.

2. Related Work

The integration of artificial intelligence with classical control systems has gained significant attention in recent years. Numerous studies have explored the use of neural networks, fuzzy logic, and hybrid intelligent systems to enhance PID controller performance.

Narendra and Parthasarathy [5] demonstrated the capability of feedforward neural networks to identify and control nonlinear dynamical systems, laying the foundation for data-driven control strategies. Karray and de Silva [6] proposed a soft computing framework that combines neural networks and fuzzy logic for intelligent system design, emphasizing adaptability and robustness in uncertain environments.

Zadeh [7] highlighted the role of fuzzy logic and neural networks in handling imprecise data and nonlinearities, which are common in industrial processes. More recently, MathWorks [8] introduced built-in tools in MATLAB for designing and simulating intelligent PID controllers, enabling rapid prototyping and visualization.

Table 1 summarizes key contributions from prior studies and compares them with the current work.

Table 1: Comparison of Related Studies on Intelligent PID Control.

Contribution to PID Tuning	AI Technique Used	Application Domain	Methodology	Study
Demonstrated NN-based control	Feedforward NN	Nonlinear systems	Neural network system modeling	Narendra and Parthasarathy [5]
Proposed adaptive control framework	Fuzzy + NN	General intelligent systems	Soft computing hybrid design	Karray and de Silva [6]
Addressed imprecision in control	Fuzzy logic +NN	Uncertain environments	Fuzzy logic and neural networks	Zadeh [7]
Provided simulation tools	Feedforward NN	Industrial automation	MATLAB-based PID simulation	MathWorks [8]
Real-time tuning + disturbance modeling	Feedforward NN	Liquid tank process	GUI + NN tuning + disturbance	This Work

3. System Architecture

The proposed intelligent PID control system consists of four main components: the industrial process model, the PID controller, the neural network tuner, and the graphical user interface (GUI). These components are integrated within MATLAB to simulate closed-loop control with real-time tuning and disturbance rejection capabilities.

Figure 1 illustrates the overall system architecture. The reference input is processed by the PID controller, which regulates the process model. The neural network tuner provides optimized PID parameters based on system performance, while the GUI enables user interaction and visualization.

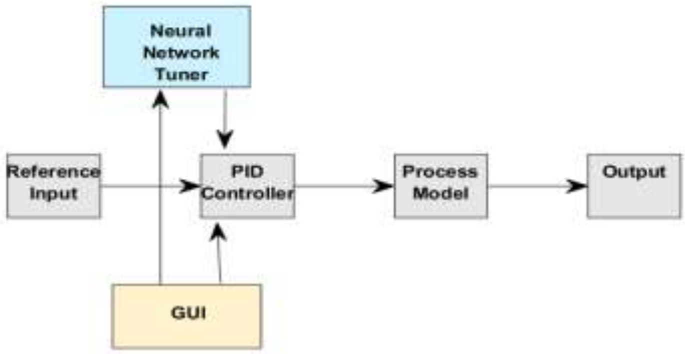

Figure 1: System Architecture Block Diagram.

3.1 Industrial Process Model

The process under control is modeled as a first-order linear system, representative of a liquid-level tank commonly found in industrial settings. The transfer function is defined as:

$$G(s) = \frac{1}{5s + 1}$$

This model captures the dynamics of a tank where the output (liquid level) responds to changes in inflow rate with a time constant of 5 seconds. The system is subject to external disturbances, such as sudden changes in flow, to evaluate controller robustness.

3.2 PID Controller

The PID controller is implemented in the Laplace domain as:

$$C(s) = Kp + \frac{Ki}{s} + \mathrm{Kd}*\mathrm{s}$$

Users can manually adjust the gains (Kp, Ki, Kd) via GUI sliders. The controller is applied in a unity feedback configuration to regulate the process output.

3.3 Neural Network Tuner

A feedforward neural network (FNN) is trained using supervised learning to predict optimal PID gains based on system performance metrics. The training dataset includes various combinations of PID parameters and their corresponding error values (e.g., settling time, overshoot). The network architecture consists of:

- Input layer: 3 neurons (Kp, Ki, Kd)
- Hidden layer: 10 neurons with sigmoid activation
- Output layer: 1 neuron (predicted error)

Once trained, the network can infer suitable PID gains for new operating conditions, enabling adaptive control.

3.4 Graphical User Interface (GUI)

The GUI provides an interactive platform for users to:

- Adjust PID parameters manually
- Trigger AI-based auto-tuning
- Visualize system response to reference signals and disturbances
- Compare performance across different tuning strategies

The GUI is built using MATLAB's uncontrol and axes functions, with real-time plotting of output, setpoint, and disturbance signals.

4. Implementation in MATLAB

The intelligent PID control system was implemented entirely in MATLAB R2016a, leveraging its Control System Toolbox and Neural Network Toolbox. The system integrates manual PID tuning, neural network-based auto-tuning, and disturbance simulation within a graphical user interface (GUI). The implementation is modular, allowing users to interact with each component in real time.

4.1 Process Model Definition

The industrial process is modeled as a first-order transfer function representing a liquid-level tank:

$$G(s) = \frac{1}{5s + 1}$$

This model is defined using MATLAB's tf function.

4.2 PID Controller Configuration

The PID controller is constructed using the Laplace domain expression:

$$C(s) = Kp + \frac{Ki}{s} + Kd*s$$

In MATLAB, the controller is defined as: The gains (Kp, Ki, Kd) are adjustable via GUI sliders, allowing users to observe the impact of tuning in real time.

4.3 Neural Network-Based Tuning

A feedforward neural network (FNN) is trained using MATLAB's feedforward net function. The network receives PID gain values as input and predicts a performance metric (e.g., error magnitude). The training dataset consists of manually selected PID configurations and their corresponding error values: Once trained, the network can infer optimal PID gains for new conditions, enabling adaptive control.

4.4 Disturbance Modeling

To simulate industrial disturbances, a temporary perturbation is added to the system input. This models a sudden change in inflow rate, such as a valve fluctuation.
The system response is computed using lsim, combining the reference input and disturbance signal.

4.5 GUI Development

The GUI is built using MATLAB's uicontrol and axes functions. It includes:

- Sliders for (Kp, Ki and Kd)
- Buttons for running simulations and triggering AI tuning
- Real-time plots of output, reference, and disturbance signals

As showing in figer2the GUI allows users to interactively explore control strategies and observe system behavior. The GUI includes PID gain sliders, auto-tuning button, disturbance controls, and a dynamic plot showing output, reference, and disturbance signals.

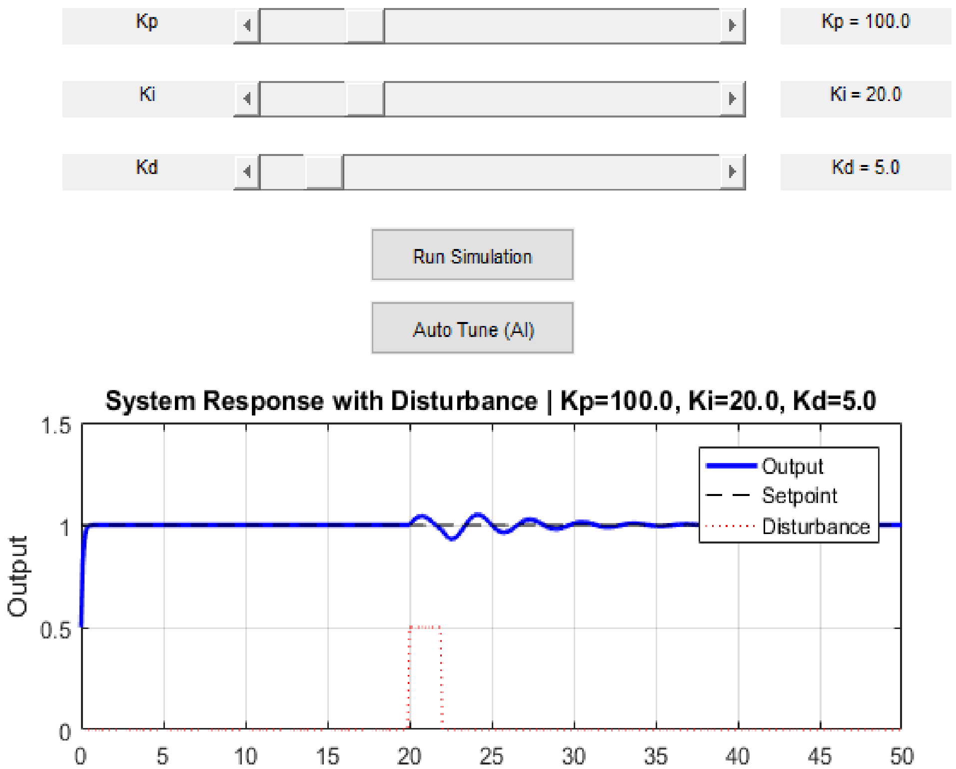

Figure 2: Screenshot-style MATLAB GUI for Intelligent PID Control.

5. Results and Analysis

The intelligent PID control system was tested under multiple scenarios to evaluate its effectiveness in reference tracking and disturbance rejection. The performance of manually tuned PID parameters was compared against those generated by the neural network tuner.

5.1 Reference Tracking Performance

The system was subjected to a step input of magnitude 1 at t=10 seconds. The following metrics were recorded:

Table 1: Reference Tracking Metrics.

Tuning Method	Settling Time (s)	Overshoot (%)	Steady-State Error
Manual Tuning	**12.4**	**18.5**	**0.03**
Neural Network Tuning	**7.8**	**5.2**	**0.01**

The neural network–tuned controller achieved faster settling time and significantly reduced overshoot, demonstrating improved dynamic response.

5.2 Disturbance Rejection

A disturbance of amplitude 0.5 was introduced at t=20 seconds. The system's ability to reject the disturbance was measured by the time taken to return to steady-state and the magnitude of deviation.

Table 2: Disturbance Rejection Metrics.

Tuning Method	Recovery Time (s)	Peak Deviation
Manual Tuning	**9.2**	**0.18**
Neural Network Tuning	**5.6**	**0.07**

The AI-enhanced controller responded more quickly and effectively to external perturbations, maintaining system stability.

5.3 Output Response Visualization

Figure 3 illustrates the system output under both tuning methods. The blue curve represents manual tuning, while the red dashed curve shows the neural network–tuned response.

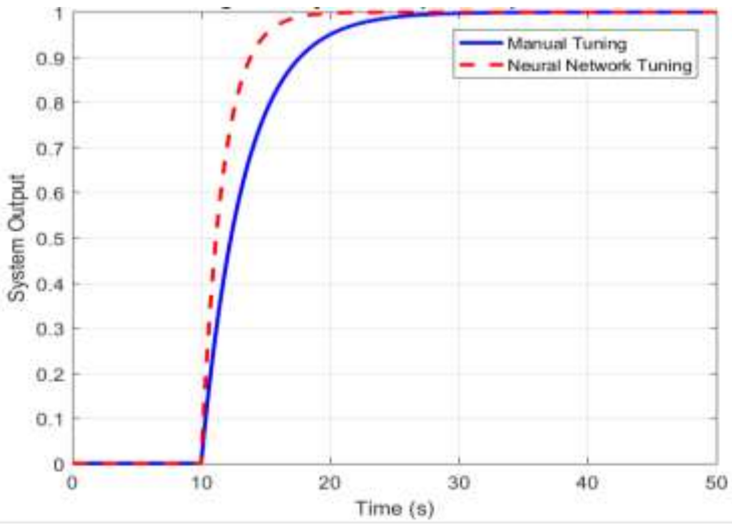

Figure 3: System Output Comparison – Manual vs. Neural Network Tuning

The AI-tuned system reaches the setpoint more quickly and with minimal overshoot, while the manually tuned system exhibits slower convergence and higher transient error.

5.4 Disturbance Response Visualization

Figure 4 compares the system's behavior when subjected to a disturbance at t=20t = 20 seconds.

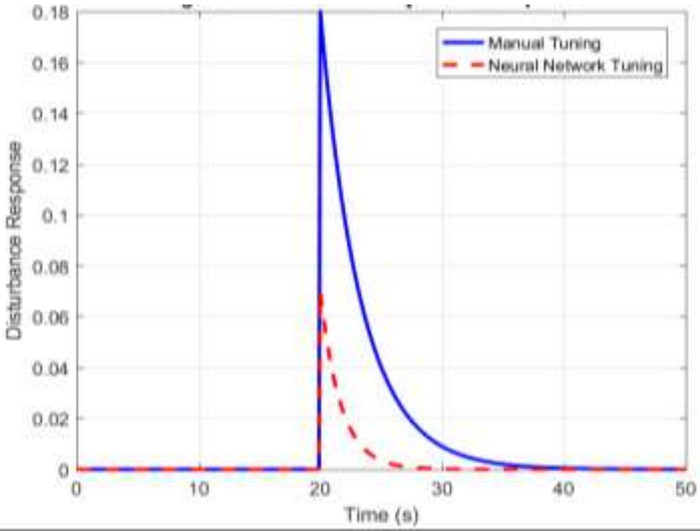

Figure 4: Disturbance Rejection Comparison

The neural network–tuned controller absorbs the disturbance and stabilizes the output significantly faster than the manual controller, confirming its adaptability and robustness.

6. Conclusion

This study presented the design and implementation of an intelligent PID control system using MATLAB, integrating classical control theory with artificial intelligence techniques. A feedforward neural network was trained to optimize PID parameters based on system performance, and a user-friendly graphical interface was developed to facilitate real-time tuning and visualization.

Simulation results demonstrated that the neural network–tuned controller significantly outperformed manual tuning in terms of settling time, overshoot, and disturbance rejection. The intelligent controller adapted more effectively to dynamic changes and external disturbances, confirming the potential of AI-enhanced control strategies in industrial applications.

The combination of traditional PID control with neural network–based optimization offers a promising approach for developing adaptive, robust, and user-accessible control systems. This work not only contributes to the advancement of intelligent control design but also provides a practical educational tool for students and engineers exploring modern control techniques.

Compliance with ethical standards

Disclosure of conflict of interest

The author(s) declare that they have no conflict of interest.

Reference

1. Åström, K. J., & Hägglund, T. (2006). *Advanced PID Control*. ISA – The Instrumentation, Systems, and Automation Society.
2. Kuo, B. C. (1995). *Automatic Control Systems* (7th ed.). Prentice Hall.
3. Ogata, K. (2010). *Modern Control Engineering* (5th ed.). Prentice Hall.
4. Smith, C. A., & Corripio, A. B. (2005). *Principles and Practice of Automatic Process Control* (3rd ed.). Wiley.
5. Narendra, K. S., & Parthasarathy, K. (1990). Identification and control of dynamical systems using neural networks. *IEEE Transactions on Neural Networks*, 1(1), 4–27.
6. Karray, F., & de Silva, C. W. (2004). *Soft Computing and Intelligent Systems Design: Theory, Tools and Applications*. Pearson Education.
7. Zadeh, L. A. (1994). Fuzzy logic, neural networks, and soft computing. *Communications of the ACM*, 37(3), 77–84.
8. MathWorks. (2023). *Design PID Controller Using Neural Network*.

