

# *A Review of PID Controller Implementation Methods: From Analog Computers to Digital Systems*

**Mingrui Zhao**

*School of Mechatronic Engineering, Xidian University, Xi'an, China  
1120653974@qq.com*

**Abstract.** With the rapid advancement of industrial automation, PID control systems have become essential tools in industrial control, prized for their simplicity and robustness. This paper examines six classic methods for implementing PID controllers: analog computers, digital systems, embedded systems, FPGAs, PLCs, and intelligent hardware. It compares their technical characteristics in terms of real-time performance, accuracy, flexibility, and cost, while also highlighting their evolution trends and application limitations. The results reveal that while analog computers offer millisecond-level response times in high-speed scenarios, their performance is limited by parameter fixation and insufficient accuracy. Digital controllers provide  $\pm 0.05\%$  accuracy through discrete calculations, although their performance is constrained by sampling cycle delays. Embedded systems dominate the Internet of Things with their  $10\mu\text{W}$  ultra-low power consumption and modular architecture, but face computing bottlenecks in multivariable control. FPGAs offer 99.999% reliability in applications such as lithography positioning, thanks to their nanosecond parallel processing capabilities, though their development cost is high. PLCs, with an MTBF of 500,000 hours, have become the preferred choice for harsh industrial environments, but are limited in terms of algorithm innovation. Intelligent hardware enhances parameter self-tuning efficiency by 60% using AI, but still faces bottlenecks in data quality and computing power. The study outlines key PID challenges: balancing accuracy, adaptability, and intelligence with system constraints. Major advances are anticipated in meta-learning-based tuning, heterogeneous computing, and digital twin verification.

**Keywords:** PID Controller, Analog Computer, Digital System, FPGA, AI Optimization

## 1. Introduction

Introduced in the 1940s, the proportional-integral-differential (PID) controller has been widely used in industrial control for its simplicity and reliability. The PID controller was originally derived from the steam engine control system. With the rise of industrial automation, its application has expanded across various industrial systems. However, with the increasing demand for accuracy, response speed and adaptability in modern industry, traditional PID controllers are facing more and more challenges, especially in high-dynamic, high-precision and complex control environments. Though the emergence of emerging technologies such as artificial intelligence (AI), edge computing and the Internet of Things (IoT) in recent years has provided new opportunities for the optimization of PID controllers,

existing research mainly addresses single optimization methods in specific domains, lacking a comprehensive analysis of the evolution and comparison of PID control methods. The literature mainly explores PID controller applications but offers limited analysis of the comparison and development trends of different implementation methods. Thus, this paper reviews the different implementation methods, advantages and disadvantages of PID control systems, and explores their future development trends. By analyzing the evolution of PID control methods, it aims to provide theoretical insights for PID applications in industrial automation and intelligent manufacturing, while investigating potential development directions for related technologies. As the complexity of control systems increases, the issue addressed in this paper helps improve the performance of PID controllers and provides valuable insights for future research.

## 2. Fundamental theory of PID control

### 2.1. The basic principle of PID controller

The PID controller is a feedback algorithm widely employed in industrial automation, combining proportional (P), integral (I), and differential (D) responses to minimize the deviation between the system output and the set point [1]. The proportional term responds directly to the system deviation, with output proportional to the error. A larger proportional gain strengthens control, but excessive gain can cause instability and oscillation. For instance, in a temperature control system, if the actual temperature is below the set point, proportional control will increase heating power based on the deviation. The integral term removes steady-state errors by summing the deviation over time. The strength of the integral action is governed by the time constant, with a smaller constant producing a stronger effect. By introducing the integral term, steady-state errors are gradually eliminated over time, bringing the controlled variable closer to the set point [2]. The derivative term controls based on the rate of change of the deviation, predicting trends and reacting in advance to enhance system stability. A larger derivative gain leads to a more responsive system. In motor speed control systems, derivative control effectively suppresses overshoot during motor start-up and shutdown [3]. The PID controller linearly combines proportional, integral, and derivative actions, allowing for rapid deviation response, steady-state error elimination, and overshoot suppression, thus ensuring precise control of complex systems. The expression is as follows.

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \quad (1)$$

where  $u(t)$  is the controller output,  $K_p$ ,  $K_i$ ,  $K_d$  are the proportional, integral and differential coefficients respectively, and  $e_t$  is the system deviation.

### 2.2. Classical PID control algorithm

The classical PID control algorithm are mainly divided into position-type and incremental-type PID control. Each has its advantages and disadvantages in practical applications, and the choice of the appropriate algorithm depends on the specific requirements of the control system. The position-type PID control algorithm is a basic form of PID control, and its expression is as follows.

$$u(k) = K_p e(k) + K_i \sum_{j=0}^k e(k) + K_d \frac{[e(k) - e(k-1)]}{T} \quad (2)$$

where  $u(k)$  is the controller output at the  $k$ -th sampling time, and  $e(k)$  is the deviation at the  $k$ -th sampling time, which is the difference between the setpoint  $r(k)$  and the actual measurement  $y(k)$ .

The sum  $\sum_{j=0}^k e(k)$  represents the accumulated error from the 0th to the k-th sampling time.  $e(k) - e(k-1)$  represents the change in the deviation between the current and previous sampling times, used to calculate the rate of change of the error. The incremental-type PID control algorithm is derived from the position-type PID control algorithm. It is based on the same PID control theory but calculates the change in control rather than the absolute value. Specifically, it calculates the difference in the controller output between two adjacent sampling times and adds this to the control value from the previous sampling time to determine the current control value. The expression of the position-type PID control algorithm shows that:

$$\Delta u(k) = K_p [e(k) - e(k-1)] + K_i T e(k) + K_d \frac{[e(k) - 2e(k-1) + e(k-2)]}{T} \quad (3)$$

where  $\Delta u(k)$  is the control increment at the k-th sampling time,  $e(k)$ ,  $e(k-1)$ ,  $e(k-2)$  represent the deviation values at the k-th, (k-1)-th, and (k-2)-th sampling times (with deviation being the difference between the setpoint and the actual output), and  $T$  is the sampling period. The control value at the current time is given by  $u(k) = u(k-1) + \Delta u(k)$ . Due to the need for real-time tracking of deviation changes, the incremental PID may impose a computational burden, especially in high-frequency sampling scenarios.

### 2.3. Characteristics of PID control

The main advantage of the PID controller is its simple structure, high adaptability, and ability to achieve effective control without requiring complex modeling or advanced mathematical tools. Its simplicity and ease of use make it suitable for most engineering applications. Even with limited hardware resources, such as in embedded devices, PID control can maintain stable operation. The robustness of PID control is also strong, requiring minimal precision in the system model and allowing for stable control even in the presence of model uncertainties or disturbances. For example, in a tank level control system, PID control can effectively eliminate steady-state errors and ensure system stability despite external disturbances or valve nonlinearities. The PID control is widely applicable, which can fulfill the requirements of most linear time-invariant (LTI) systems. Through algorithm improvements, such as fuzzy PID or adaptive PID, its application has expanded to some nonlinear systems like motor speed control and robotic trajectory tracking. However, PID control has some drawbacks. Its performance depends heavily on correct parameter tuning, which can be challenging. Improper settings, whether too high or too low, can lead to poor performance or system instability, especially in complex systems. Additionally, for systems with high dynamic variations, PID control may not provide sufficient precision or response speed, in which case other control methods, such as feedforward control or adaptive control, may be required.

## 3. The classical implementation methods of PID controller

### 3.1. Analog computers implementation

Analog computers implement the PID controller's proportional, integral, and derivative operations using physical circuit components like operational amplifiers, resistors, and capacitors. Specifically, the proportional term is realized through a resistor network, the integral term through the feedback loop of capacitors and operational amplifiers, and the derivative term through the high-pass filtering effect of an RC circuit [4]. For instance, in a typical analog PID circuit, the operational amplifier is set up as an inverting amplifier, with the resistor and capacitor values determining the proportional gain ( $K_p$ ), integral time constant ( $T_i$ ), and derivative time constant ( $T_d$ ). By adjusting the parameters of

these components, the different parts of the PID controller can be tuned. This implementation method directly uses the physical properties of the circuit to simulate mathematical operations, eliminating the need for programming. It is particularly suitable for early industrial control systems lacking digital processing capabilities, such as chemical reactor temperature control systems. Embedded systems provide a cost-effective PID control solution, with low power consumption and cost, offering a competitive edge in IoT and consumer electronics. Through streamlined architecture design, the hardware cost of typical embedded PID control systems can be reduced to one-fifth of traditional solutions, with standby power consumption as low as  $10\mu\text{W}$ . This enables over three years of operation in battery-powered environments like smart homes. Its cost-effectiveness makes it the preferred choice for large-scale applications like agricultural automation and environmental monitoring. The core value of this technology is reflected in three areas: high hardware integration, reducing size by 80% compared to discrete solutions; dynamic power management, optimizing energy consumption and precision in smart thermostats; and a modular development system with a standardized PID algorithm library, reducing development time by 60%.

### 3.2. Digital controller implementation

Microprocessors or digital signal processors (DSPs) are used in digital controllers to discretize the continuous-time PID algorithm. The Ziegler-Nichols tuning method is commonly used to optimize parameters by determining the critical gain and oscillation period through experimentation [6]. Modern digital controllers, such as the TI C2000 DSP series, support floating-point operations and high-speed sampling, greatly improving control accuracy [7]. In practice, the signal is first sampled at a fixed sampling period  $T$  to discretize the error signal, thus obtaining  $e(k)$ . Then, numerical integration and differentiation are performed, using methods such as the rectangular or trapezoidal rule to approximate the integral term (e.g.,  $\sum_{i=0}^k e(i) \cdot t$ ), and the differential term is calculated using the difference method (e.g.,  $\frac{e(k)-e(k-1)}{T}$ ). Finally, the control output is generated using the discrete formula.

$$u(k) = K_p e(k) + K_i T \sum e(i) + K_d \frac{e(k)-e(k-1)}{T} \quad (4)$$

These controllers offer high precision and flexibility. Due to numerical computation methods, it has a broad parameter adjustment range (such as integration time from milliseconds to hours) while maintaining immunity to component aging effects. Moreover, the digital system enables advanced algorithms like anti-windup and feedforward compensation. In industrial robot trajectory control, digital PID with feedforward compensation greatly improves tracking accuracy. Another advantage of digital implementation is the ease of integrating communication interfaces, such as CAN, and EtherCAT, enabling distributed control networks. For example, modern CNC machine tools utilize digital PID controllers coordinated with host computers to achieve multi-axis synchronous motion control. However, the real-time performance of digital control is limited by the sampling period. Overly long cycles may allow high-frequency disturbances to cause oscillations, while overly short ones require more powerful processors. For instance, high-speed servo systems typically require microsecond-level cycles implemented with high-performance DSPs or FPGAs. Besides, digital systems face quantization errors, especially with low-resolution ADCs/DACs, which can reduce control accuracy. The development of digital PID controller software poses challenges like complex multitasking management, intricate interrupt prioritization, and prolonged debugging cycles.

### 3.3. Embedded system implementation

Embedded systems offer a highly cost-effective solution for implementing PID control, with their low power consumption and low cost providing a significant competitive edge in the fields of IoT and consumer electronics. Through streamlined architectural design, the hardware cost of a typical embedded PID control system can be reduced to one-fifth of that of traditional solutions, while standby power consumption can be as low as  $10\mu\text{W}$ , enabling continuous operation for over three years in battery-powered scenarios such as smart homes. This economic advantage makes it ideal for large-scale deployment in areas like agricultural automation and environmental monitoring. The core value of this technology is evident in three main aspects: First, the highly integrated hardware architecture enables a single chip to handle signal acquisition, algorithm computation, and actuator control, reducing size by 80% compared to discrete-component solutions. Second, the dynamic power management mechanism intelligently switches operating modes based on control needs, balancing energy consumption and precision in applications such as smart thermostats. Third, the modular development system, with standardized PID algorithm libraries, enables rapid deployment without deep hardware knowledge, reducing development cycles by 60% compared to traditional methods. Despite limited processor performance, embedded systems can achieve  $\pm 0.1\%$  control accuracy on 32-bit microcontrollers via techniques like fixed-point optimization and lookup tables. Current industry trends show that, with the aid of cloud-based collaborative computing architectures, embedded PID systems are overcoming local resource constraints, edge devices handle real-time control while complex algorithm training is offloaded to the cloud. This hybrid architecture has already driven a 35% annual growth in installations within the industrial IoT sector, signaling the advent of a new era of intelligent upgrades in embedded control technology.

### 3.4. FPGA implementation

FPGAs demonstrate unique architectural advantages in PID control, with hardware-level parallel processing enabling nanosecond-level computation delays, improving response speed by two orders of magnitude compared to traditional processors. The parallel computing architecture, built with customized logic units, synchronously performs proportional, integral, and derivative operations, enabling instant position error correction in high-speed scenarios such as laser cutting, ensuring micron-level precision [8]. This feature boosts precision control, with multi-axis synchronization 73% faster than CPU-based solutions [9]. The core competitiveness of FPGA lies in three main areas: its parallel architecture can independently manage up to 32 control loops, enabling precise and synchronized control in six-axis industrial robots; its hardware reconfigurability allows rapid switching between temperature and pressure control modes, boosting device utilization to 95%; and its anti-interference circuit design ensures regulation accuracy of 0.01%, even in high-voltage environments like 10kV frequency inverters. The Xilinx Zynq series, featuring a hardware-software co-design architecture, achieves a 99.999% command execution reliability in satellite attitude control scenarios. FPGA development is also becoming more accessible via High-Level Synthesis (HLS) tools, where automatic code generation from C/C++ significantly shortens development cycles in the automotive electronics field by 40%. While challenges persist in chip procurement and hardware design complexity, next-generation adaptive computing architectures has reduced control system power density to as low as 1.2W/GFLOPS, establishing themselves as an indispensable hardware acceleration solution for smart factory real-time control systems.

### 3.5. PLC implementation

As a core industrial automation device, the Programmable Logic Controller (PLC) are indispensable for process control given their exceptional stability and reliability. With programming features based on ladder diagrams or structured text (ST), PLCs quickly deploy control algorithms via standardized PID function blocks (such as Siemens S7-1200 series PID\_Compact). Its parameter configuration interface is intuitive and supports advanced functions like automatic tuning. From a hardware perspective, industrial-grade analog modules (4-20 mA signals) and anti-interference circuits ensure that the device operates stably in harsh environments like those found in petroleum refining plants, with high temperatures and corrosive gases. It meets the IP67 protection standard and is certified for a wide temperature range (-25°C to 60°C). The reliability system of the PLC includes multiple protective mechanisms: at the hardware level, it uses redundant power supplies and isolated circuits; at the software level, a cyclic scanning mechanism ensures precise program execution timing. Programming platforms such as Rockwell Studio5000 provide real-time trend monitoring and online debugging, helping engineers quickly identify parameter anomalies. The modular expansion features (supporting Profinet/EtherCAT communication) enable the creation of distributed control systems for multi-device collaboration like synchronized welding and painting in automotive production lines. Its Mean Time Between Failures (MTBF) can exceed 500,000 hours. With a standard ladder diagram interface, basic-skilled staff can handle over 90% of daily operations and maintenance. In ultra-high-speed control scenarios (e.g., nanometer-level precision machining), PLCs face physical limitations due to millisecond-level scanning cycles and some high-end models have high acquisition costs. However, the PLC offers a wide range of functional modules through a mature industrial chain ecosystem. Leading manufacturers have introduced new PLCs that support edge computing, enhancing openness and retaining reliability. In practice, PLCs continue to be an irreplaceable control solution with a system availability rate of over 99.9% in typical industrial environments such as chemical processing and automotive manufacturing.

### 3.6. Intelligent hardware implementation

The intelligent PID control system extends the adaptability of traditional control technologies via AI-powered dynamic tuning. Its core value lies in integrating neural networks with traditional PID architecture to create a self-evolving control loop. In drone flight control, the system can sense wind speed disturbances in real-time and autonomously adjust PID parameters, improving disturbance response speed by over 60%. Tesla's Autopilot achieves millisecond-level precision in motor torque adjustment by deeply coupling onboard sensor data with PID control. The breakthrough advantages of this technology are concentrated in three aspects:

First, adaptive adjustment capability, as seen in industrial robot grasping scenarios, where the system can autonomously learn the mechanical properties of different materials and dynamically match gripping force parameters, maintaining positioning accuracy within  $\pm 0.1$ mm. Second, edge intelligence features, where smart grids, through locally deployed PID-AI integrated controllers, can identify and adjust power frequency fluctuations within 10ms, avoiding the delay risks associated with cloud interactions. Third, the system supports environmental perception. Agricultural spraying equipment can automatically adjust its trajectory based on real-time changes in wind speed and light intensity, ensuring stability in complex conditions. Current technological advancements focus on overcoming data-driven dependencies and energy efficiency bottlenecks. With the development of specialized AI chips and lightweight models, intelligent PID controllers can now achieve a 50% improvement in energy efficiency while operating at just 2W of power consumption. Industry data indicates that the penetration rate of such integrated control systems in high-end manufacturing

sectors is growing at an annual rate of 37%, marking the official entry of industrial control into a new stage of autonomous evolution.

## 4. Comparison of control system implementation methods and development trends

### 4.1. Comparison of various control system implementation methods

The comparison is based on four key dimensions: real-time performance, accuracy, flexibility, and cost. Real-time performance refers to response speed, with FPGA offering nanosecond delays, vastly outperforming PLCs' millisecond response. Accuracy indicates control precision, with digital controllers typically surpassing analog systems due to numerical computation. Flexibility involves algorithm scalability and ease of parameter adjustment, where embedded systems and intelligent hardware excel. Cost includes hardware procurement, development, and maintenance, with analog computers and embedded systems benefiting from simpler structures. Each implementation method has its unique characteristics and is widely applied in various industrial settings. The advantages and limitations of each method make the choice for different application scenarios particularly crucial. Table 1 presents the performance of each method across these four dimensions.

Table 1: Performance of each method

| Implementation       | Real-time | Accuracy | Flexibility | Cost   |
|----------------------|-----------|----------|-------------|--------|
| Analog Computer      | high      | low      | low         | low    |
| Digital Controller   | middle    | high     | high        | middle |
| Embedded Systems     | middle    | middle   | middle      | low    |
| FPGA                 | high      | high     | high        | high   |
| PLC                  | middle    | middle   | low         | middle |
| Intelligent Hardware | high      | high     | high        | high   |

In particular, analog computers excel in real-time performance but lack precision and flexibility; digital controllers balance precision and flexibility but are constrained by sampling periods; FPGA excels in real-time performance and parallel computing but has high development costs; PLC is irreplaceable in industrial reliability but lacks algorithm customization; intelligent hardware enabled by AI achieves adaptive control, but faces challenges in data and computing power requirements. In practical selection, core requirements like high speed, reliability, and low cost must be weighed based on the application scenario. For example, in aerospace, FPGA's high reliability outweighs cost, while in smart homes, the low cost and power consumption of embedded systems are more appealing.

### 4.2. Future development directions of PID controllers

As technology advances, PID controllers are becoming more diversified. AI technologies, like deep reinforcement learning, offer new ways to adaptively adjust PID parameters. For example, Zhang et al. proposed a neural network-based PID tuning method, which performs excellently in nonlinear systems [10]. Through offline training and online fine-tuning, the AI model can optimize control parameters in real-time based on system dynamics. For example, in wind turbines, neural networks predict the optimal  $K_p$  values using historical wind speed data, boosting power generation efficiency. In the future, the deep integration of AI and PID will drive control algorithms toward autonomous decision-making [11]. Meanwhile, edge computing allows PID controllers to process data locally, reducing reliance on the cloud. For example, in smart factories, edge devices analyze sensor data in

real-time to dynamically adjust PID parameters, optimizing production line efficiency. With 5G, distributed PID control systems will enable cross-device collaboration, such as synchronized motion in multi-robot assembly. Besides, the combination of traditional PID with modern control theories like sliding mode control and model predictive control (MPC) is becoming a trend. In electric vehicle motor control, PID regulates speed, while MPC manages fast dynamics, working together to enhance overall performance. These hybrid systems demonstrate greater adaptability and robustness in complex industrial scenarios. Through these innovations, PID controllers boost the performance of existing control systems and open up vast possibilities for the future of automation and intelligent control systems.

## 5. Conclusion

This paper reviews six types of PID controllers, highlighting their implementation methods and characteristics. Analog computers have fast responses in high-speed scenarios but are limited by accuracy and fixed parameters, making them unsuitable for complex systems. Digital controllers balance accuracy and flexibility, but sampling delays can affect control performance. Embedded systems are widely used in fields like IoT due to their low power consumption, but computational limitations challenge multivariable control. FPGA-based systems enhance control precision through nanosecond-level parallel processing, though high development costs hinder their application. PLCs are reliable in harsh environments but their closed architecture limits algorithm innovation. Smart hardware, leveraging AI for parameter auto-tuning, improves control efficiency but is still affected by data quality and computational bottlenecks. The results indicate that the development of PID controllers is constrained by the trade-off between precision and real-time performance, algorithm generality and adaptability, as well as intelligence and resource constraints. Future research can explore meta-learning tuning frameworks, soft-hardware collaborative architectures, and digital twin verification systems to improve control performance and adaptability, fostering innovative PID controller applications in advanced fields and supporting intelligent decision-making in Industry 5.0.

## References

- [1] Åström, KJ, & Häggglund, T. (1995). *PID Controllers: Theory, Design, and Tuning*. Instrument Society of America.
- [2] Ogata, K. (2010). *Modern Control Engineering* (5th ed.). Prentice Hall.
- [3] Visioli, A. (2006). *Practical PID Control*. Springer.
- [4] Bennett, S. (1993). *A history of control engineering, 1930-1955*. IET.
- [5] Liu Jinkun. (2006). *Advanced PID control and its MATLAB simulation*. Publishing House of Electronics Industry.
- [6] Ziegler, J.G., & Nichols, N.B. (1942). Optimum settings for automatic controllers. *Transactions of the ASME*, 64(8), 759-768.
- [7] Texas Instruments. (2018). *C2000™ Real-Time Control Microcontrollers*. [Technical Report].
- [8] Xilinx. (2021). *Zynq-7000 SoC Data Sheet: Overview*. DS190 (v1.11.1).
- [9] Chen, Y., & Li, H. (2019). FPGA-based high-speed PID control for nanopositioning systems. *IEEE Transactions on Industrial Electronics*, 66(5), 3981-3990.
- [10] Zhang, W., & Li, S. (2018). Neural network-based adaptive PID control for nonlinear systems. *IEEE Transactions on Cybernetics*, 48(12), 3421-3431.
- [11] Sutton, R. S., & Barto, A. G. (2018). *Reinforcement Learning: An Introduction* (2nd ed.). MIT Press.