KFUPM
Journal of Undergraduate Research International

Diamond Open Access 54

Article Computer Engineering/Control and Instrumentation Engineering

# Hybrid Fixed-Point Control Architecture for Quadrotor Stabilization Using FOPI/FOPID on FPGA

Abdullah Nader Alkhater[1] and Ghulam E Mustafa Abro[2*]

Cite https://doi.org/10.64589/juri/209726

**Submitted:** May 19, 2025 **Revised:** July 21, 2025 **Accepted:** August 20, 2025

**ABSTRACT**

Maintaining stability in underactuated systems (e.g., quadrotor aerial vehicles (QAVs)) presents significant control challenges in aerial robotics. This study conducts a simulation-based performance evaluation of fractional-order Proportional Integral (FOPI) and fractional-order Proportional Integral derivative (FOPID) controllers designed for potential deployment on an FPGA-based platform. A hybrid fixed-point/floating-point architecture was developed in MATLAB/Simulink to optimize computational precision, memory utilization, and processing speed. The control algorithms were validated through simulations and hardware-in-the-loop testing using Xilinx Vivado to emulate real-time Field Programmable Gate Array (FPGA) behavior. Although the control architecture has not yet been physically implemented on FPGA hardware, the simulation framework closely mirrors deployment conditions. The results indicate that the FOPID controller achieves superior accuracy, response time, and resource efficiency compared to traditional designs. This simulation-driven analysis highlights the feasibility of FPGA-based real-time controllers for QAVs applications, laying the groundwork for future physical implementations and flight validations.

**Keywords:** FPGA, quadrotor UAV, fractional order PID, Xilinx Vivado, and simulation-driven

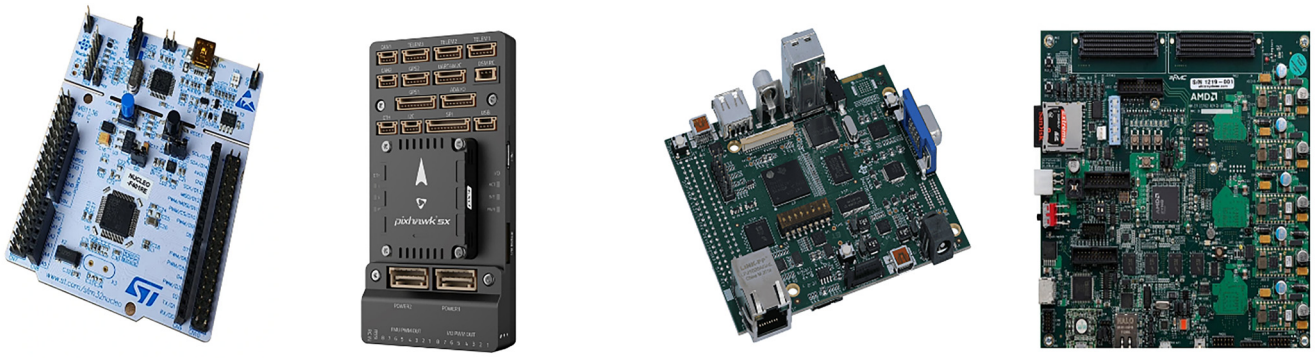
## 1. INTRODUCTION

In aerospace, surveillance, delivery, and autonomous systems, achieving precise control, stability, and real-time responsiveness is essential. Consequently, unmanned aerial vehicles (UAVs) are increasingly deployed in complex and dynamic environments. UAV control faces considerable challenges, including nonlinear dynamics, external disturbances, underactuated behavior, and constraints in onboard processing capabilities [1,2]. To address these issues, advanced control systems must ensure energy efficiency, adaptability, and resilience to disturbances and faults. Pixhawk controllers are widely used in UAVs due to their cost-effectiveness, compatibility with PX4 and ArduPilot, and user-friendliness [3,4]. However, their microcontroller-based sequential architecture limits real-time processing, computational efficiency, and scalability of advanced control algorithms [4].

The limitations of Pixhawk become particularly evident in high-frequency control applications, where delays in trajectory tracking and system responses pose significant challenges. Conversely, field-programmable gate arrays (FPGAs) present a compelling alternative, offering features such as concurrent task execution, improved accuracy through floating-point operations, and real-time responsiveness with reduced latency [5]. FPGAs facilitate energy-efficient and fault-tolerant designs by incorporating dynamic reconfiguration capabilities that address hardware or computational anomalies [6–11]. These benefits make FPGA-based systems ideal for complex UAV control scenarios that demand enhanced responsiveness, reliability, and energy efficiency. As UAV missions increase in complexity, the requirement for high-performance real-time control solutions underscores the superiority of FPGA-based designs over traditional Pixhawk systems [12–16]. This study performs a simulation-based analysis of FPGA-driven control techniques for quadrotor aerial vehicles (QAVs) using MATLAB and Simulink to model, evaluate, and compare advanced controllers within a realistic computational framework. Although the solution has not been implemented on actual hardware, the simulated models and HIL-based configurations mimic real-time FPGA deployment to evaluate system behavior and performance trends. The key contributions of this simulation-based study are as follows:

- Development of a hybrid fixed-point and floating-point processing method within the MATLAB/Simulink simulation environment, designed for FPGA platform implementation and optimized for memory efficiency, computational speed, and power simulation modeling.
- The simulation results indicate that FOPID control provides greater precision, faster response, and enhanced resource efficiency compared to conventional controllers, underscoring its suitability for real-time implementation in FPGA-based aerial robot systems.

The remainder of this paper is organized into five sections. Section 1 introduces the subject and outlines the research



(a) STM32 Controller (b) Pixhawk Controller (c) TMDSLCDK6748 DSP Board (d) AMD Zynq™ 7000

**Figure 1.** Modern Processors Utilised as Flight Controllers for Unmanned Aerial Vehicles

contributions. Section 2 reviews state-of-the-art controllers and their respective limitations. Section 3 elaborates on the proposed control algorithms. Section 4 presents the simulation results. Section 5 offers conclusions, along with future recommendations and directions.

## 2. STATE-OF-THE-ART CONTROLLER BOARDS

When comparing various UAV control platforms—such as STM32-, Pixhawk-, Digital Signal Processor (DSP)-, and FPGA-based systems (Fig. 1)—each exhibits unique strengths and weaknesses. Pixhawk is a widely adopted open-source flight controller that is cost-effective and compatible with PX4 firmware, and it provides a flexible platform for diverse UAV applications. Nonetheless, its performance is constrained by limited memory and lower computational speed, particularly when implementing advanced control algorithms like model predictive control (MPC), leading to compromised real-time responsiveness[17]. STM32-based controllers are noted for their compact size and economical design. They typically support proportional integral derivative (PID) control and Kalman filter-based sensor fusion, facilitating swift response times. However, their limited processing power restricts the implementation of computationally demanding control schemes[18]. Digital signal processors (DSPs) offer rapid processing and real-time responsiveness, making them suitable for applications requiring precise control. Despite these benefits, DSPs may entail higher costs, complex hardware integration, and decreased flexibility when modifying control algorithms[19].

In contrast, FPGA controllers are superior choices for high-performance UAVs. Their capacity for concurrent task execution facilitates the simultaneous processing of multiple control loops, thereby diminishing latency. In contrast to microcontroller-based systems like STM32, Pixhawk, or DSPs, FPGAs allow dynamic hardware reconfiguration, enabling real-time adaptation or refinement of control algorithms without the limitations of fixed architectures. This adaptability renders FPGAs exceptionally suitable for implementing intricate, computation-intensive control strategies such as MPC, reinforcement learning, and real-time adaptive control[19]. Notably, Pixhawk boards are esteemed for their energy efficiency and sensor modularity, whereas contemporary low-power FPGA platforms (e.g., 28 nm or 16 nm series) are increasingly competitive in power consumption through methodologies like clock gating and dynamic reconfiguration. Furthermore, FPGAs support modular interfacing through protocols such as SPI, I2C, and UART, thus facilitating integration with a wide range of sensors. Although this requires more custom development compared to the native support of Pixhawk, it offers unmatched flexibility and hardware-level optimization tailored to specific mission requirements.

## 3. CONTROL ALGORITHMS

**3.1. Proportional Integral and Derivative (PID) Control.** The proportional–integral–derivative (PID) controller is among the most extensively employed techniques in control systems. It consists of three components: proportional (P), integral (I), and derivative (D), each contributing to the reduction of system errors and the enhancement of stability[20–23]. The PID controller is represented in the Laplace domain by the following transfer function:

$$\frac{U(s)}{E(s)} = K_p + \frac{K_i}{s} + K_d s \tag{1}$$

In the equation above, $K_p$ is the proportional gain that modifies the output based on the current error. $K_i$ is the integral gain addresses accumulated past errors to eliminate steady-state error. Finally, $K_d$ represents the derivative gain that responds to the rate of error change, thereby reducing overshoot and enhancing stability[22]. Unlike the standard PID controller, Fractional-Order Proportional-Integral (FOPI) and Fractional-Order Proportional-Integral-Derivative (FOPID) controllers expand the conventional model by incorporating non-integer (fractional) orders for the integral and derivative terms, thereby providing enhanced flexibility and tuning precision. The transfer function of the FOPI controller in the Laplace domain is expressed as follows:

$$G_{FOPI}(s) = K_p + \frac{K_i}{s^{\lambda}} \tag{2}$$

where $\lambda \epsilon [0,1]$ represents the fractional order of integration. The FOPID controller further generalizes the PID structure as

$$G_{FOPID}(s) = K_p + \frac{K_i}{s^{\lambda}} + K_d s^{\mu} \tag{3}$$

where $\lambda$, $\mu \epsilon [0,1]$ denotes the fractional integration and differentiation orders are defined, respectively. These fractional-order elements enable finer control dynamics, particularly advantageous for systems with complex nonlinearities, such as quadrotors. In this study, FOPI and FOPID designs were implemented and optimized within a MATLAB/Simulink framework for deployment on FPGA hardware.

**3.2. FPGA-enabled FOPID Control.** The study employed VHDL and Xilinx tools to realize FOPI and FOPID controllers on an FPGA platform. FPGAs are versatile, programmable circuits that serve as robust alternatives to fixed hardware solutions like ASICs and SoCs. Several boards utilize a 28 nm logic architecture, offering improved power efficiency and performance compared to conventional systems. They comprise 6.6 million logic cells and high-speed transceivers (up to 12.5 GB/s), making them suitable for real-time control applications.

**3.3. Design and Implementation of a PI-PID Controller Based on FPGAs.** The design process began with formulating a discrete FOPID controller derived from its continuous counterparts. The transformation was executed in MATLAB, and the discrete FOPID model was realized in the Z-domain[5–9]. To optimize FOPID parameters for FPGA implementation, we employed MATLAB's Fixed-Point Designer Toolbox to determine optimal bit allocations for each controller component. Our design assigns 25 bits to the fractional part and 7 bits to the integer part. Additionally, we utilize a floating-point approach adhering to the IEEE 754 double-precision standard, ensured by Model Advisor Toolbox of MATLAB/Simulink. We then generated hardware description language (HDL) code via high-level synthesis (HLS) and integrated it into the system model. Controller performance was evaluated using oscilloscope simulations and Xilinx viewers with the Vitis Composer tool, followed by an assessment of system energy and resource usage. This process culminated in the development and validation of an FPGA-based FOPI-FOPID controller, demonstrating its effectiveness and efficiency in real-time control applications.

**Table 1.** *Summary related to Xilinx FPGA Utilization*

| Logic Utilization | Available | Used | Utilization |
|---|---|---|---|
| I/O Pins | 200 | 92 | 46% |
| Logic LUTs | 40500 | 502 | 1.23% |
| Slice LUTs | 40500 | 502 | 1.23% |
| Slice Registers | 81100 | 56 | 0.069% |

## 4. RESULTS AND DISCUSSIONS

The discrete-time transfer function modeling the pitch dynamics of the quadrotor, derived by discretizing the linearized continuous-time system with a zero-order hold and sampling period $T_s = 0.05\ s$, is expressed as follows:

$$G(z) = \frac{b_1 z + b_0}{z^2 + a_1 z + a_0}. \tag{4}$$

Furthermore, it can be articulated as

$$G(z) = \frac{0.1875z + 0.0055}{z^2 - 1.974z + 0.814}, \tag{5}$$

where the coefficients $b_1, b_0$, $a_1$, $a_0$ arise from the inertia of the quadrotor, damping effects, and the selected sampling rate. This second-order transfer function captures the essential pitch response to control inputs and serves as the foundation for the subsequent performance and stability analysis of the digital controller. The selected parameters for the FOPI/FOPID controller are: $K_p = 0.5137$, $K_i = 0.5137$, and $K_d = 0.788$. The Fixed-Point Designer Toolbox in MATLAB/Simulink was employed to determine optimal bit allocation for effectively implementing the controller in a fixed-point format. The setup assigned 25 bits to the fractional component and 7 bits to the integer component. A data type conversion (DTC) method was integrated into the model to transform real-world numerical values into fixed-point format via appropriate scaling. This scaling preserves a balance between numerical precision and value range, thereby reducing issues such as overflow or saturation. The fixed-point controller was assessed across various simulated scenarios to confirm its stability and control performance. Compatibility evaluations were performed using the Model Advisor Toolbox to ensure the controller satisfied all necessary implementation criteria.

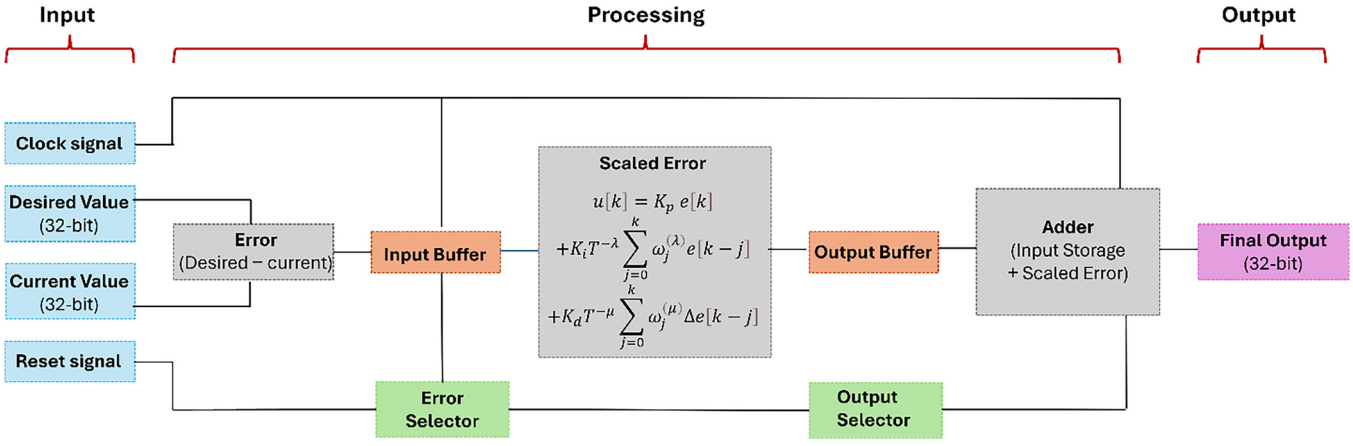



**Figure 2.** Block diagram of RTL components

**Table 2.** *Summary related to Xilinx FPGA Utilization*

| No of Inputs | Bits | Adders | Muxes |
|---|---|---|---|
| 2-input | 28 | 1 | 1 |
| 2-input | 26 | 1 | 1 |
| 2-input | 27 | 1 | 1 |
| 2-input | 24 | 1 | 1 |
| 2-input | 23 | 1 | 1 |
| 3-input | 8 | 2 | 10 |
| 2-input | 8 | 2 | 3 |
| 6-input | 5 | - | 3 |
| 3-input | 5 | - | 3 |
| 4-input | 5 | - | 3 |
| 2-input | 1 | - | 40 |

The circuit design demonstrates the use of a 32-bit hybrid fixed-and-floating-point approach utilizing FPGA resources. Table 1 details the primary FPGA resource utilization, with input pins specifically allocated and distributed among bitwise adders and multiplexers (MUXes). The XOR gate was the sole gate utilizing a 2-input configuration, efficiently using designated bits. Additionally, Figure 2 depicts the FPGA resource distribution for MUX, CARRY logic, OBUF (Output Buffer), and IBUF (Input Buffer), highlighting the distinct contributions of each register-level component to the overall hardware implementation. The OBUF transmitted the output signal to external pads, while the IBUF handled single-ended signals. Table 2 offers a comprehensive overview of the input distribution. The allocation of logic slices and lookup tables (LUTs) was carefully organized to optimize hardware efficiency.

Figures 3 and 4 offer a comprehensive evaluation of the FPGA-based controller, which enhances model execution through a hybrid fixed-and-floating-point approach to improve computational precision and stability.

The substantial difference in power values between Figures 3 and 4 arises from different analysis scenarios. Specifically, Figure 3 illustrates high-frequency synthesis-level dynamic activity, representing peak operating power, while Figure 4 shows post-implementation power after optimizations like clock gating and resource sharing under low-load conditions. Figure 5

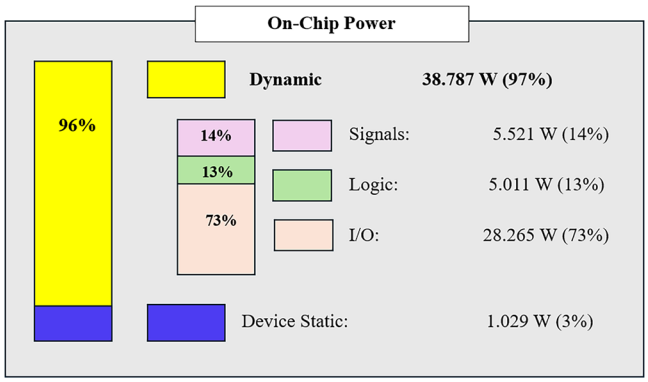



**Figure 3.** Summary Related to power consumption with 38.787W

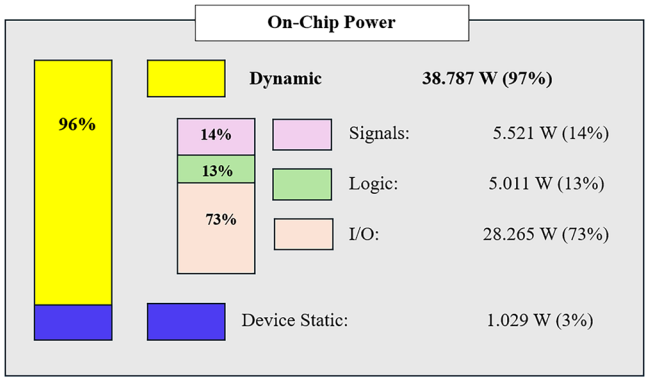



**Figure 4.** Summary Related to power consumption with 0.015W

assesses the performance of the FOPI/FOPID controller following a 25-s disturbance. A step-response comparison of FOPI and FOPID controllers applied to a quadrotor pitch control system under a disturbance introduced at 25 s demonstrates the superior dynamic performance of FOPID controller, including faster settling time, reduced overshoot, and enhanced disturbance rejection compared to the FOPI controller. Additionally, the analysis revealed that the PID controller achieved quicker settling times and less overshoot than the PI controller.

Table 3 summarizes the comparative rise and settling times of the PI and PID controllers. The PI controller exhibited a rise time of 1.71 s, whereas the PID controller demonstrated a more favorable rise time of 1.32 s. Furthermore, the PID controller achieved a superior settling time of 38.52 s with an overshoot of 0.95%, indicating outstanding performance. Notably, the controller gains used in this study were determined through an iterative simulation-based tuning process in MATLAB. Initial values were established using standard tuning heuristics and subsequently refined to optimize key performance metrics—rise time, settling time, and overshoot—for the linearized pitch dynamics of the quadrotor. The optimized parameters ensured stable, responsive control while meeting fixed-point implementation constraints for FPGA deployment.

The results in Figure 5 and Table 3 illustrate the superior dynamic performance of the FOPID controller over the FOPI controller. In addition to improved rise and settling times, the

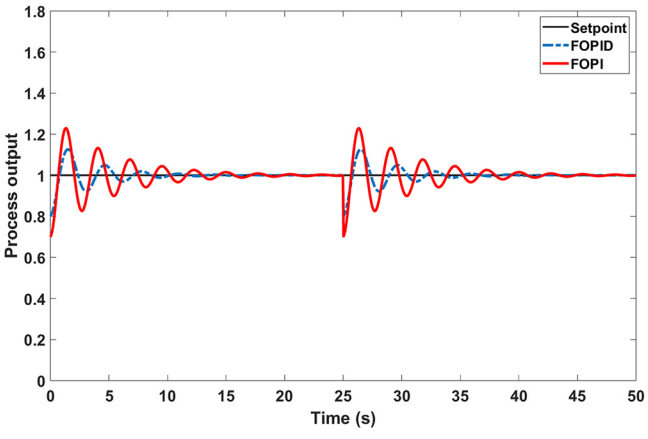



**Figure 5.** Comparative Performance of FOPI and FOPID Control Schemes

**Table 3.** *Performance Analysis of Proposed Control Schemes*

| Control Algorithms | Settling time $T_s$ | Rise time $T_r$ | Overshoot (%OS) |
|---|---|---|---|
| PI | 40.12 | 1.71 | 1.21 |
| PID | 38.52 | 1.32 | 0.95 |

FOPID controller demonstrated enhanced disturbance robustness, characterized by quicker recovery and reduced overshoot following a step disturbance at 25 s. This highlights the derivative component in the FOPID structure's role in enhancing damping and predictive action, enabling more effective responses to transient deviations. Such performance is particularly valuable in UAV applications, where rapid and stable reactions to external perturbations are critical. These findings validate the effectiveness of the proposed control scheme in realistic flight scenarios and emphasize its potential for real-time FPGA implementation.

## 5. CONCLUSIONS

This study presents a simulation-based evaluation of FPGA-implemented FOPI and FOPID control techniques for QAVs, harnessing the computational advantages of FPGA platforms. The hybrid fixed-point and floating-point methodology, developed within MATLAB/Simulink and synthesized via Xilinx Vivado, facilitated high-performance real-time control with enhanced precision and energy efficiency. A comparative study demonstrated that the FPGA-based FOPID controller outperformed the conventional FOPI method in tracking performance, disturbance rejection, and settling time, all while utilizing fewer hardware resources. These results validate the integration of advanced control algorithms with reconfigurable FPGA architectures to enhance the stability and responsiveness of aerial robotic systems. A balanced approach between numerical precision and hardware efficiency was achieved through a hybrid fixed- and floating-point control architecture. By employing fixed-point representation alongside floating-point operations in critical computational blocks, the proposed design attained real-time performance suitable for FPGA deployment in UAVs, where resource optimization is essential. This hybrid strategy provides a practical balance that circumvents the limitations of purely fixed-point or fully floating-point implementations.

## 6. FUTURE DIRECTIONS AND RECOMMENDATIONS

To further this research, several future directions are suggested, including implementation on an FPGA board, real-time drone maneuvering, and the incorporation of wireless telemetry and cloud-based control diagnostics for comprehensive system monitoring and remote upgrades. Future studies should investigate swarm coordination methodologies and fault-tolerant control mechanisms across multiple UAVs utilizing distributed FPGAs through actual hardware deployment. These enhancements would improve system flexibility and scalability, expanding the applicability of the proposed approach to various real-world autonomous aerial applications. Additionally, we intend to extend the current methodology by integrating nonlinear quadrotor dynamics, external disturbances, and parametric uncertainties to more accurately simulate real-world conditions. The existing linearized model has been expanded to a full six-degrees-of-freedom representation to assess its robustness and stability under aggressive flight scenarios. Moreover, we aim to evaluate the proposed FOPI/FOPID implementation against advanced control strategies, including MPC, sliding mode control, and adaptive control techniques. These comparisons will provide a comprehensive understanding of the scalability and practical applicability of FPGA-based control architectures for real-time UAV operations.

## AFFILIATIONS AND AUTHOR DETAILS


### Undergraduate Author

**Abdullah Nader Alkhater** – *Department of Aerospace Engineering, King Fahd University of Petroleum & Minerals, Saudi Arabia;* 0009-0004-2186-3742
Email: s202161390@kfupm.edu.sa

### Corresponding Author

**Ghulam E Mustafa Abro** – *Research Mendor, Interdisciplinary Research Centre for Aviation and Space Exploration, King Fahd University of Petroleum & Minerals, Saudi Arabia;* 0000-0003-1874-1889
Email: Ghulam.abro@kfupm.edu.sa


## ACKNOWLEDGEMENTS

The authors acknowledge the support provided by the Department of Aerospace Engineering and the Interdisciplinary Research Center for Aviation and Space Exploration at King Fahd University of Petroleum and Minerals. This research was funded by the Undergraduate Research Office (URO) KFUPM, Saudi Arabia.

## REFERENCES


(1) M. A. Fawwaz, K. Bingi, R. Ibrahim, P. A. M. Devan, and B. R. Prusty, "Design of PIDD controller for robust performance of process plants," Algorithms, vol. 16, no. 9, p. 437, 2023.

(2) A. A. Aguirre, L. D. Munoz, C. A. Martın, M. J. Ramırez, and C. A. Salazar, "Design of digital PID controllers relying on FPGA-based techniques," IFAC-PapersOnLine, vol. 51, no. 4, pp. 936–941, 2018.

(3) J. Lima, R. Menotti, J. M. Cardoso, and E. Marques, "A methodology to design FPGA-based PID controllers," in Proc. IEEE Int. Conf. Systems, Man and Cybernetics (SMC), vol. 3, pp. 2577–2583, 2006.

(4) Y. Xu, K. Shuang, S. Jiang, and X. Wu, "FPGA implementation of a best-precision fixed-point digital PID controller," in Proc. Int. Conf. Measuring Technology and Mechatronics Automation, vol. 3, pp.384–387, 2009.

(5) Y. F. Chan, M. Moallem, and W. Wang, "Design and implementation of modular FPGA-based PID controllers," IEEE Trans. Ind. Electron., vol. 54, no. 4, pp. 1898–1906, 2007.

(6) B. Sreenivasappa and R. Udaykumar, "Design and implementation of FPGA-based low power digital PID controllers," in Proc. Int. Conf.Industrial and Information Systems (ICIIS), pp. 568–573, 2009.

(7) S. Chander, P. M. Agarwal, and I. Gupta, "FPGA-based PID controller for DC-DC converter," in Proc. Joint Int. Conf. Power Electronics, Drives and Energy Systems & Power India, pp. 1–6, 2010.

(8) L. F. Castano and G. A. Osorio, "Design of an FPGA-based position PI servo controller for a DC motor with dry friction," in Proc. VII Southern Conf. Programmable Logic (SPL), pp. 75–80, 2011.

(9) Ş. Akkaya, O. Akbatı , and H. Gorgün, "Multiple closed-loop system control with digital PID controller using FPGA," in Proc. Int. Conf. Control, Decision and Information Technologies (CoDIT), pp. 764–769, 2014.

(10) P. Chotikunnan and R. Chotikunnan, "Dual design PID controller for robotic manipulator application," J. Robot. Control (JRC), vol. 4, no. 1, pp. 23–34, 2023.

(11) A. Kumar and R. Phadke, "Design of digital PID controller for blood glucose monitoring system," Int. J. Eng. Res. Technol., vol. 3, no. 12, pp. 307–311, 2014.

(12) A. Ali, K. Bingi, R. Ibrahim, P. A. M. Devan, and K. Devika, "A review on FPGA implementation of fractional-order systems and PID controllers," AEU-Int. J. Electron. Commun., p. 155218, 2024.

(13) B. Jayakrishna and V. Agarwal, "FPGA implementation of QFT-based controller for a buck-type DC-DC power converter and comparison with fractional and integral order PID controllers," in Proc. IEEE Conf. Industrial Electronics and Applications, 2008.

(14) F. Zhang and Z. Li, "Design of fractional PID control system for BLD motor based on FPGA," in Proc. Chinese Control and Decision Conf. (CCDC), 2018.

(15) H. Khati, A. Fekik, A. T. Azar, M. A. Nehmar, et al., "Optimizing UAV stability and control with FPGA-based PID control system design," in Proc. Int. Conf. Control, Automation, and Diagnosis (ICCAD), 2024.

(16) A. Ali, R. Ibrahim, K. Bingi, et al., "Hybrid fixed- and floating-point approach for implementing PID controllers on DE1-SoC FPGA," in Proc. Symp. Computer Applications & Industrial Electronics (ISCAIE), 2024.

(17) S. Amadi, "Design and implementation of Pixhawk-based control systems for Quadrotor UAVs," Int. J. Aerosp. Eng., 2018. [Online]. Available: https://doi.org/10.1109/CESYS.2016.7889898.

(18) H. Cai, Z. Wu, and M. Chen, "Design of STM32-based Quadrotor UAV control system," KSII Trans. Internet Inf. Syst., vol. 17, no. 2, pp. 353–360, 2023. [Online]. Available: https://doi.org/10.3837/tiis.2023.02.004.

(19) P. Liu, Z. Li, and C. Cui, "Design of unmanned aerial vehicle (UAV) flight control system based on DSP and adaptive control theory," in Proc. Int. Conf. Communication and Electronics Systems (ICCES), pp. 1–5, 2016.

(20) J. Lima, R. Menotti, J. M. Cardoso, and E. Marques, "A methodology to design FPGA-based PID controllers," in Proc. IEEE Int. Conf. Systems, Man and Cybernetics (SMC), vol. 3, pp. 2577–2583, 2006.

(21) Y. Xu, K. Shuang, S. Jiang, and X. Wu, "FPGA implementation of a best-precision fixed-point digital PID controller," in Proc. Int. Conf. Measuring Technology and Mechatronics Automation, vol. 3, pp. 384–387, 2009.

(22) Y. F. Chan, M. Moallem, and W. Wang, "Design and implementation of modular FPGA-based PID controllers," IEEE Trans. Ind. Electron., vol. 54, no. 4, pp. 1898–1906, 2007.

(23) B. Sreenivasappa and R. Udaykumar, "Design and implementation of FPGA-based low power digital PID controllers," in Proc. Int. Conf. Industrial and Information Systems (ICIIS), pp. 568–573, 2009.