Journal of Engineering Research (Libya)

COMPARATIVE PERFORMANCE EVALUATION OF CLASSICAL, OPTIMAL, AND INTELLIGENT CONTROLLERS FOR REAL-TIME EMBEDDED MODULAR MULTILEVEL CONVERTER SYSTEMS

Nada A. Alhamdi[*] and Abdelmuhaimen Albasir[**]

[*]Department of Electrical and Computer Engineering, Faculty of Engineering, Wayne State University, Detroit, USA
[**]Department of Electrical and Electronics Engineering, Faculty of Engineering, University of Tripoli, Libya
Email: alhamdi@wayne.edu
Received 9 August 2026; Revised 28 September 2026; Accepted 30 September 2026; Published 4 October 2026.

تقييم الأداء المقارن لوحدات التحكم التقليدية، المثلى، والذكية لأنظمة المحولات المعيارية متعددة المستويات المضمنة في الوقت الفعلي

ندى الحامدي[*]، عبد المهيمن البصير[**]

[*]قسم الهندسة الكهربائية والحاسب، كلية الهندسة، جامعة واين ستيت، الولايات المتحدة الأمريكية
[**]قسم الهندسة الكهربائية والالكترونية، كلية الهندسة، جامعة طرابلس، ليبيا

الملخص

يتطلب التحكم المضمَّن في الزمن الفعلي للمحولات المعيارية متعددة المستويات (MMCs) كلاً من أداء تحكم عالٍ وكفاءة حسابية مرتفعة، وذلك بسبب الديناميكيات المترابطة للمحول ومتطلبات موازنة الطاقة المعقدة. تقارن هذه الورقة بين ثلاث فئات من المتحكمات :متحكم التناسبي–التكاملي–التفاضلي (PID) التقليدي، والمنظم التربيعي الخطي الأمثل (LQR) ذو تأثير تكاملي ، ومتحكم المنطق الضبابي الذكي(FLC) ، وذلك باستخدام نموذج موحّد لمحول معياري متعدد المستويات (MMC) ثلاثي الطور. يتم تقييم المتحكمات من حيث زمن الصعود ، وزمن الاستقرار، والتجاوز، وخطأ الحالة المستقرة، والمتانة، والتعقيد الحسابي، وزمن تأخير التنفيذ، ومتطلبات الذاكرة، ومدى ملاءمتها للتنفيذ في الزمن الحقيقي. تُظهر عمليات المحاكاة باستخدام MATLAB/Simulink أن متحكم LQR يوفر أداءً عابرًا وتخميدًا أفضل، بينما يوفر متحكم FLC متانة عالية في مواجهة اللاخطية وعدم اليقين في المعلمات، في حين يتميز متحكم PID بأقل عبء حسابي وأبسط تنفيذ ضمن الأنظمة المضمنة. وتوفر النتائج إرشادات عملية لاختيار استراتيجيات التحكم المناسبة للمحولات المعيارية متعددة المستويات(MMC) ، مع الأخذ في الاعتبار كلًا من الأداء الديناميكي وقيود الحوسبة المضمنة.

الكلمات المفتاحية: التحكم المضمَّن في الزمن الفعلي، المحول المعياري متعدد المستويات (MMC)، المنظم التربيعي الخطي(LQR) ، المتحكم التناسبي–التكاملي–التفاضلي(PID) ، المتحكم بالمنطق الضبابي(FLC) .

Copyright © 2026 by Faculty of Engineering, University of Tripoli, Libya.
This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

ABSTRACT

Real-time embedded control of Modular Multilevel Converters (MMCs) requires both high control performance and efficient computation due to the converter's coupled dynamics and complex energy-balancing requirements. This paper compares three controller classes a classical Proportional–Integral–Derivative (PID) controller, an optimal Linear Quadratic Regulator (LQR) with integral action, and an intelligent Fuzzy Logic Controller (FLC)—using a common three-phase MMC model. The controllers are evaluated in terms of rise time, settling time, overshoot, steady-state error, robustness, computational complexity, execution latency, memory requirements, and real-time implementation suitability. MATLAB/Simulink simulations show that LQR provides superior transient performance and damping, FLC offers strong robustness to nonlinearities and parameter uncertainty, and PID provides the lowest computational burden and simplest embedded implementation. The results provide practical guidance for selecting MMC control strategies according to both dynamic performance and embedded computing constraints.

KEYWORDS: Real-time embedded control, Modular Multilevel Converter (MMC), Proportional–Integral–Derivative (PID) controller, Linear Quadratic Regulator (LQR), Fuzzy Logic Controller (FLC).

INTRODUCTION

The rapid evolution of embedded computing technologies has significantly transformed the design of modern control systems. Contemporary industrial infrastructures increasingly rely on real-time embedded controllers capable of performing high-speed sensing, computation, decision-making, and actuation within strict timing constraints. These controllers constitute the computational core of numerous cyber-physical systems, including renewable energy systems, smart grids, electric transportation, autonomous vehicles, industrial automation, and intelligent manufacturing. As computational requirements continue to increase, controller design is no longer evaluated solely according to dynamic performance but also according to execution efficiency, memory utilization, scalability, and real-time implementation capability.

Among modern power conversion technologies, MMCs have emerged as one of the most important converter topologies for medium- and high-voltage applications because of their modular structure, excellent harmonic performance, scalability, and high conversion efficiency [1]. These characteristics have led to widespread deployment of MMCs in High Voltage Direct Current (HVDC) transmission systems, renewable energy integration, Flexible AC Transmission Systems (FACTS), large battery energy storage systems, and high-power motor drives [2]. However, these advantages are accompanied by substantially increased computational complexity due to the large number of switching devices, capacitor voltage states, circulating currents, and energy-balancing variables that must be simultaneously regulated [3], [4].

The increasing complexity of MMC dynamics creates a challenging real-time control problem. As the number of converter sub-modules increases, the controller must continuously process a growing number of state variables while maintaining system stability under strict sampling deadlines [5]. Consequently, the selection of an appropriate control strategy directly influences not only transient performance but also

processor utilization, execution latency, memory consumption, and overall embedded system reliability. Classical control methods, particularly the PID controller, continue to dominate industrial applications because of their straightforward implementation, low computational requirements, and minimal memory footprint. Nevertheless, decentralized PID control often struggles to compensate for the strong coupling among MMC state variables, leading to degraded transient response and reduced robustness under parameter variations [6]. Optimal control approaches, including the LQR, address many of these limitations by exploiting complete state feedback to simultaneously regulate multiple interacting variables. Intelligent control techniques, particularly FLC, further extend controller capabilities by accommodating nonlinear behavior and parameter uncertainty without requiring an exact mathematical model of the converter [7], [8]. Although numerous studies have investigated PID, LQR, and FLCs individually for MMC applications, existing research primarily emphasizes electrical performance metrics such as current regulation, capacitor voltage balancing, disturbance rejection, and transient response.

As embedded processors increasingly serve as the computational backbone of modern converter control systems, these implementation characteristics have become critical factors in controller selection. Accordingly, this paper extends the conventional comparative analysis of MMC controllers by integrating real-time embedded systems evaluation into the performance assessment framework. A common state-space representation of a three-phase MMC is employed to design three representative controller classes: a classical PID controller, an optimal LQR controller with integral action, and an intelligent FLC. Beyond conventional transient-response evaluation, each controller is analyzed with respect to computational complexity, execution latency, implementation scalability, processor resource utilization, and suitability for deployment on embedded computing platforms operating under hard real-time constraints.

RELATED WORK

Early research on MMC control primarily emphasized converter modeling and current regulation. Wang et al. [4] presented one of the earliest comprehensive surveys on MMC modeling, modulation, and control, demonstrating that the increasing number of sub-modules significantly enlarges the system state dimension and consequently increases controller complexity. Similarly, Marquardt [5] discussed future challenges associated with large-scale MMC implementation, emphasizing the need for more advanced control techniques capable of maintaining stability while handling increasing computational requirements. Ang et al. [6] provided a comprehensive review of PID control technology, demonstrating its widespread industrial adoption across numerous automation applications. Münch et al. [7] proposed an integrated current control and energy-balancing strategy based on state-space control, demonstrating improved regulation compared with decentralized approaches.

More recently, Jouybary et al. [8] enhanced LQR performance by formulating controller synthesis as a Linear Matrix Inequality (LMI) optimization problem capable of handling parameter uncertainty. In MMC applications, fuzzy controllers utilize heuristic knowledge to regulate converter variables based on tracking error and error variation, enabling improved tolerance to nonlinearities, parameter uncertainty, and modeling inaccuracies. Lesnicar and Marquardt [9] introduced the modular multilevel

converter topology, establishing the foundation for high-voltage modular power conversion. Subsequently, Akagi [10] presented a comprehensive classification of modular multilevel converters and highlighted their suitability for medium- and high-power applications because of their modularity, scalability, and improved harmonic performance.

The evolution of embedded processors has fundamentally changed the implementation of advanced control algorithms. Modern MMC controllers are increasingly deployed on Digital Signal Processors (DSPs), ARM Cortex microcontrollers, Field-Programmable Gate Arrays (FPGAs) [11], and heterogeneous embedded computing platforms capable of executing sophisticated control algorithms within microsecond-level sampling intervals.

Table 1: Comparison of Existing MMC Control Studies.

Ref.	Controller	Main Contribution	Limitation	Embedded Analysis
[1]	LQR	Optimal state-feedback with prescribed stability	Requires accurate model	No
[4]	Survey	Comprehensive MMC modeling and control review	No controller comparison	No
[6]	PID	Classical industrial controller	Limited MIMO capability	Limited
[7]	LQR	Integrated current and energy control	High computational complexity	No
[8]	LQR (LMI)	Robust optimal control under uncertainty	Offline optimization	No
This Work	PID, LQR, FLC	Comparative evaluation of control performance and embedded implementation metrics	—	Yes

The literature demonstrates that PID, LQR, and FLCs each provide distinct advantages for MMC regulation. Classical PID controllers offer low computational cost and straightforward implementation but may exhibit degraded performance in highly coupled systems. Optimal LQR controllers provide excellent dynamic response through multivariable state feedback but require accurate mathematical models and increased computational resources. FLCs improve robustness against nonlinearities and parameter uncertainty but introduce additional computational overhead associated with fuzzy inference. Accordingly, this work addresses this gap by presenting a unified comparative framework in which PID, LQR, and Fuzzy Logic controllers are evaluated using both conventional control-performance metrics and embedded systems performance indicators as mentioned in Table (1). By combining dynamic response analysis with computational complexity and real-time implementation considerations, the proposed study provides practical guidance for selecting controller architectures suitable for next-generation embedded cyber-physical energy systems.

SYSTEM MODELLING

This section presents the mathematical model of the (MMC) adopted as the common computational plant for evaluating the three control strategies considered in this study.

A. Modular Multilevel Converter Structure

The general topology of a three-phase MMC consists of six arms, two per phase of upper and lower, each composed of a series connection of identical sub-modules. Each sub-module contains an energy storage capacitor that can be inserted into, or bypassed from, the corresponding arm, thereby synthesizing the desired output voltage waveform [12]. Arm inductors are placed between the upper and lower arms to limit transient and circulating currents and to provide the required filtering. For the purpose of this study, an average-value model of the MMC is adopted, in which each arm is represented by an equivalent controllable voltage source, consistent with the modeling approach commonly used for MMC control design.

B. AC Current Dynamics

For each phase, the AC-side output current, is defined as the difference between the upper-arm current i_p and the lower-arm current in:

$$i_s = i_p - i_n \tag{1}$$

while the internal circulating current i_d is defined as half of the sum of the two arm currents:

$$i_d = \frac{ip + in}{2} \tag{2}$$

Applying Kirchhoff's voltage law to each phase and transforming the resulting three-phase equations into the synchronously rotating dq-reference frame, the AC current dynamics can be expressed as:

$$\frac{d}{dt} i_{s,q} = -\omega\, i_{s,d} - \frac{R}{L} i_{s,q} - \frac{1}{L} v_{eac,q} - \frac{2}{L} v_{s,q} \tag{3}$$

where R and L denote the equivalent arm resistance and inductance, ω is the grid angular frequency, v_{eac} is the AC component of the converter's internal control voltage, and v_c is the grid-side voltage.

Similarly, the circulating (differential) current dynamics for each phase can be written as:

$$\frac{d}{dt} i_d = -\frac{R}{L} i_d - \frac{1}{L} v_i + \frac{1}{L} V_{DC} \tag{4}$$

where v_i represents the internal control signal associated with the circulating current, V_{DC} is the DC-link voltage, R: Equivalent resistance , i_p: Upper-arm current, i_n: Lower-arm current, i_d :Internal circuit current.

C. Energy Balancing Dynamics

Beyond current regulation, an equally critical control objective in MMC operation is maintaining the balance of stored energy across the sub-modules. Three levels of energy balancing are typically considered:

The total stored energy per phase, e_t, is governed by the sum of the upper- and lower-arm instantaneous power:

$$\frac{d}{dt}\, e_t \;=\; \frac{is}{2} v_e \;+\; i_d v_i \tag{5}$$

Differential Energy Balancing: To maintain balance between the upper and lower arms of the same phase, the differential energy, e_d, is defined based on the difference between upper- and lower-arm power:

$$\frac{d}{dt}\, e_d \;=\; \frac{is}{2} v_i \;+\; id v_e \tag{6}$$

Individual Module Balancing: For converters with N sub-modules per arm, an additional control signal is required to balance the energy of each individual module relative to its share of the total arm energy:

$$v_{pj} \;=\; \frac{vp}{N} \;+\; \Delta v_{pj}\,,\; j \;=\; 1, \ldots, N \tag{7}$$

where Δv_{pj} represents the corrective control signal for module j, subject to the constraint that the sum of all corrective signals within an arm equals zero.

D. Embedded Computational Model

Unlike conventional MMC studies that terminate with the mathematical formulation, the proposed work considers the state-space equations as the computational core of a real-time embedded controller. During each sampling interval, the embedded processor sequentially performs the following operations:

1. Acquire converter current and voltage measurements using Analog-to-Digital Converters (ADCs).
2. Update the state vector using the discrete-time state-space model.
3. Execute the selected control algorithm (PID, LQR, or Fuzzy Logic).
4. Generate updated modulation references.
5. Produce Pulse Width Modulation (PWM) signals for converter switching.
6. Repeat the process at the next sampling instant.

This embedded execution sequence establishes a direct connection between the mathematical model and the practical implementation of real-time digital control systems. It also enables evaluation of computational complexity, execution latency, and processor utilization in addition to conventional control-performance metrics [13].

REAL-TIME EMBEDDED CONTROL ARCHITECURE

Unlike conventional MMC control studies that focus primarily on control algorithms, this work considers the complete real-time embedded control architecture responsible for executing the controller within strict timing constraints as shown in Figure (1). Modern Modular Multilevel Converters are controlled by embedded computing platforms such as DSPs, ARM Cortex microcontrollers, FPGAs, or System-on-Chip (SoC) devices, where sensing, computation, and actuation must be completed within each sampling period [14]. The proposed architecture provides a unified framework for evaluating PID, LQR, and Fuzzy Logic controllers under identical embedded implementation conditions. The proposed embedded control system consists of six functional modules that execute sequentially during every sampling interval. During each control cycle, analog current and voltage measurements are acquired through high-speed Analog-to-Digital Converters. The sampled measurements are

processed by a state-estimation module that computes the variables required by the selected controller. Depending on the experimental scenario, one of the three controllers (PID, LQR, or Fuzzy Logic) generates updated modulation commands, which are subsequently converted into Pulse Width Modulation (PWM) signals that drive the converter switches. The resulting converter output is continuously measured and fed back to the controller, thereby forming a closed-loop embedded control system.

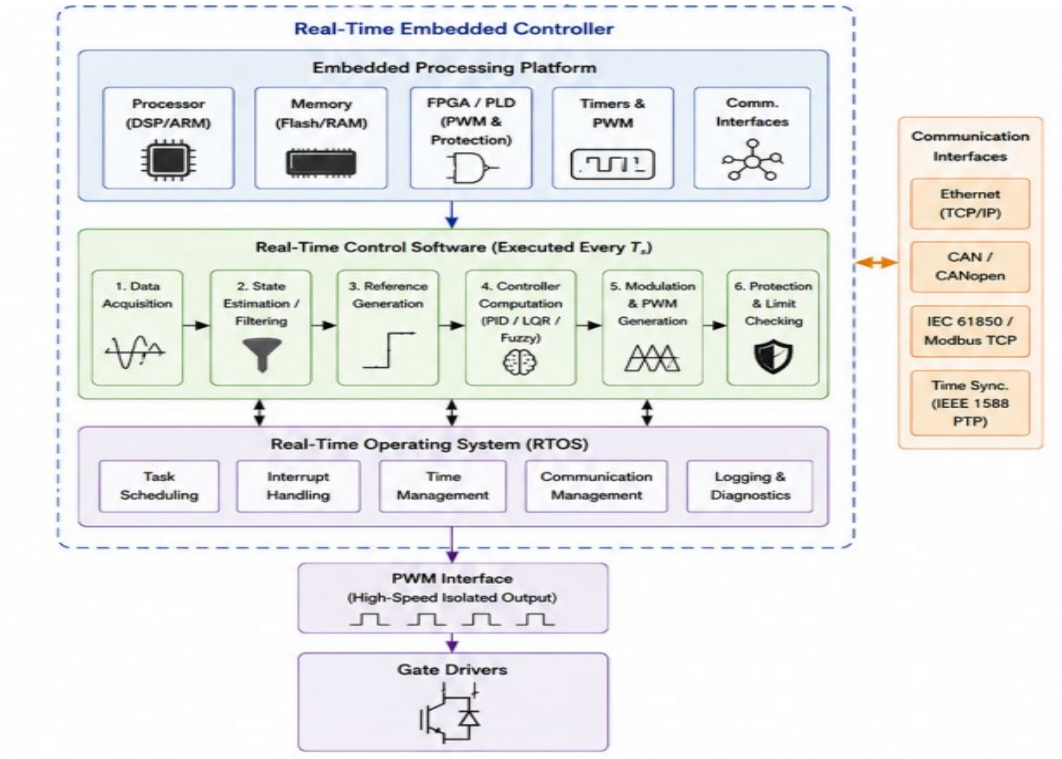

Figure 1: Real-Time Embedded Control Architecture for MMCs.

CONTROLLER DESIGN

This section presents the detailed design procedure of the three control strategies considered in this study: the LQR with integral action, the classical PID controller, and the FLC.

A. Linear Quadratic Regulator (LQR) with Integral Action

The MMC is inherently a multi-input multi-output (MIMO) system, in which several state variables including the direct- and quadrature-axis currents, the circulating current, and multiple energy states must be regulated simultaneously. Classical decentralized controllers often fail to account for the coupling between these state variables, which can lead to degraded dynamic performance or instability under parameter variation. The LQR technique provides a systematic, optimization-based framework for designing a full-state feedback controller that explicitly accounts for the interactions between all state variables, while offering an explicit trade-off between control effort and state regulation performance through the choice of weighting matrices. The LQR control law is formulated as a full-state feedback law:

$$u = -Kx \tag{8}$$

where K is the state feedback gain matrix, computed such that the closed-loop system minimizes the following quadratic performance index:

$$J = \int_0^\infty (x^T Q x + u^T R u)\, dt \tag{9}$$

Here, Q is a positive semi-definite matrix that penalizes deviations of the state variables from their reference values, and R is a positive-definite matrix that penalizes excessive control effort.

Assuming the standard linear time-invariant state-space representation of the system, $\dot{x}=Ax+Bu$, where A is the system matrix and B is the input matrix, the optimal gain matrix K is given by:

$$K = R^{-1}B^T P \tag{10}$$

where P is the unique positive-definite solution of the algebraic Riccati equation:

$$A^T P + PA - PBR^{-1}B^T P + Q = 0 \tag{11}$$

A pure state-feedback LQR controller, as formulated above, does not guarantee zero steady-state error in the presence of a constant reference or a constant disturbance, since it lacks integral action. To address this limitation, the state vector is augmented with the integral of the tracking error of the regulated outputs:

$$x_{aug} = [x \ \int (r - y)\, dt\,] \tag{12}$$

where r is the reference vector and $y = C_1 x$ is the vector of measured/controlled outputs. The augmented system dynamics can then be written as:

$$A_I = [A, 0\,;\, -C_1, 0],\ \ B_I = [B\,;\, 0]$$

The augmented control law becomes:

$$u = -K_{states}x - K_i \int (r - y)\, dt \tag{13}$$

where the augmented gain matrix $K_{aug} = [K_{states}\ K_i]$ is obtained by solving the Riccati equation associated with the augmented system:

$$K_{aug} = R^{-1}{B_I}^T P \tag{14}$$

This augmented formulation ensures that the closed-loop system tracks step-type reference changes in the AC current and energy states with zero steady-state error, which is essential for practical MMC operation.

B. PID Controller Design

Despite the growing prevalence of advanced control techniques, the classical PID controller remains one of the most widely adopted control structures in industrial power electronic applications, owing to its structural simplicity, intuitive tuning process, and low computational burden [6].

For the MMC system under study, four independent PID control loops are designed, corresponding to the four principal regulated quantities identified in the state-space model: the direct-axis AC current, is, d, the quadrature-axis AC current, is, q, the total energy per phase, et, and the differential energy between the upper and lower arms, ed. Each loop is closed independently using a standard PID control law, with the coupling terms between the $d-q$ axes (arising from the ω cross-coupling terms in (3))

treated as external disturbances that the integral action of each loop is expected to reject in steady state.

For a generic controlled variable $y(t)$ with reference $r(t)$ and tracking error $e(t) = r(t) - y(t)$, the PID control law is defined as:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)\, d\tau + K_d \frac{de(t)}{dt} \quad (15)$$

where K_p, K_i, and K_d are the proportional, integral, and derivative gains, respectively.

For practical implementation, particularly in the presence of switching-induced high-frequency noise inherent to power converter measurements, a filtered derivative term is employed:

$$C(s) = K_p + \frac{Ki}{s} + \frac{Kd\, s}{1 + s/N} \quad (16)$$

where N is the derivative filter coefficient, selected to attenuate high-frequency noise amplification while preserving the phase-lead benefit of the derivative term within the controller's effective bandwidth.

The PI gains (the derivative term is typically omitted for current loops due to the already fast, well-damped nature of a first-order plant and the risk of derivative-induced noise amplification) are tuned using the internal model control (IMC) or pole-placement method, such that the desired closed-loop bandwidth ωc is achieved:

$$K_p = \omega_c L, \quad K_i = \omega_c R \quad (17)$$

The gains for the energy balancing loops are initially estimated using the Ziegler–Nichols frequency-response method, in which the ultimate gain Ku and ultimate oscillation period Tu are determined from the plant's response at the stability boundary, and are subsequently fine-tuned in simulation to meet the following empirical design criteria:

$$K_p = 0.6K_u, \quad K_i = \frac{1.2\, Ku}{Tu}, \quad K_d = 0.075\, K_u T_u \quad (18)$$

Because the control outputs of the PID loops (the modulation index components) are subject to hard physical saturation limits imposed by the converter's maximum achievable voltage, integrator windup is a significant concern, particularly during large reference step changes or fault conditions. To mitigate this, a clamping anti-windup scheme is implemented for each PID loop, in which the integral term accumulation is halted whenever the controller output saturates and the sign of the error would further drive the output into saturation:

$$\frac{d}{dt}(\int e\, dt) = \{e(t) \quad if\ u\ not\ saturated, \quad 0 \quad otherwise \quad (19)$$

C. Fuzzy Logic Controller (FLC) Design

FLC offers a fundamentally different design philosophy: rather than relying on an explicit mathematical model of the plant, it encodes control action through a set of linguistic rules that emulate human-like reasoning about the relationship between system error and the appropriate corrective control action. Each FLC follows consisting of four sequential stages: fuzzification, rule base evaluation, inference, and defuzzification. For each control loop, two input variables are defined: the tracking error

$e(t) = e(t) - \mathrm{y(t)}$, and the rate of change of the error, $\dot{e}(t) = \frac{de(t)}{dt}$ and one output variable is defined: the incremental control action, $\Delta u(t)$ which is accumulated over time to produce the total control signal:

$$u(t) = u(t-1) + \Delta u(t) \tag{20}$$

Both input variables, e and ė, and the output variable Δu, are normalized to a common universe of discourse [−1, 1] using appropriate scaling gains (Ge, Gė, and Gu, respectively), which are tuned during the simulation stage to match the physical range of each specific control loop. Each normalized variable is partitioned into five fuzzy sets, using triangular membership functions with 50% overlap between adjacent sets, labeled as {NB, NS, ZE, PS, PB}, corresponding to Negative Big, Negative Small, Zero, Positive Small, and Positive Big, respectively. The fuzzy rule base consists of 25 rules (5×5 combinations of the two input variables), following the standard heuristic structure commonly used in fuzzy PI-type controllers. Each rule takes the general form:

$$IF\ e\ is\ A_i\ AND\ \dot{e}\ is\ B_j\ THEN\ \Delta u\ is\ C_{ij} \tag{21}$$

where $Ai, Bj \in \{NB, NS, ZE, PS, PB\}$ and Cij is determined according to a rule table (Table 1) constructed following the standard diagonal heuristic: when both the error and its rate of change are large and of the same sign, a large corrective action is applied; when the error is near zero and decreasing in magnitude, the corrective action is reduced to avoid overshoot; and when the error and its rate of change have opposite signs (indicating the system is already correcting itself), a smaller or even opposite corrective action is applied to avoid oscillatory behavior.

For real-time implementation, the fuzzy inference mechanism described above is typically realized either through direct rule evaluation at each sampling instant or through a pre-computed look-up Table (2) relating the discretized (e, ė) input space to the corresponding Δu output, the latter being computationally more efficient and therefore more suitable for the sampling rates required in power converter control applications.

Table 2: Fuzzy Rule Base for Δu (Rows: e, Columns: ė).

e \ ė	NB	NS	ZE	PS	PB
NB	NB	NB	NB	NS	ZE
NS	NB	NB	NS	ZE	PS
ZE	NB	NS	ZE	PS	PB
PS	NS	ZE	PS	PB	PB
PB	ZE	PS	PB	PB	PB

COMPARATIVE PERFORMANCE EVALUATION

This section presents a comprehensive comparative evaluation of the three controllers. Unlike traditional MMC studies that evaluate controller performance solely using electrical performance indices, all controllers are implemented using the same nonlinear average-value MMC model and executed within the identical real-time embedded architecture.

A. Simulation Environment

The three controllers PID, LQR with integral action, and FLC are implemented and evaluated on a nonlinear average-value model of a three-phase MMC in MATLAB/Simulink. All three controllers regulate the same four quantities ($i_{s,d}, i_{s,q}$, $, e_t, e_d$) of the same plant, under identical reference and disturbance scenarios, ensuring a fair basis for comparison. The simulation parameters used for the study are summarized in Table (3). The PID controller is well-tuned. The inner Current Loop is fast enough to catch faults, the outer Energy Loop is slow enough to avoid fighting the inner loop, and both have enough 'safety margins' to handle real-world changes (like temperature or load) without becoming unstable as shown in Figure (2).

Table 3: MMC Parameters.

Parameter	Value
Rated power, S	100 MVA
DC-link voltage, VDC	80 kV
Arm inductance, L	15 mH
Arm resistance, R	0.2 Ω
Sub-modules per arm, N	40
Sub-module capacitance	10 mF
Fundamental frequency	50 Hz
AC grid voltage	33 kv

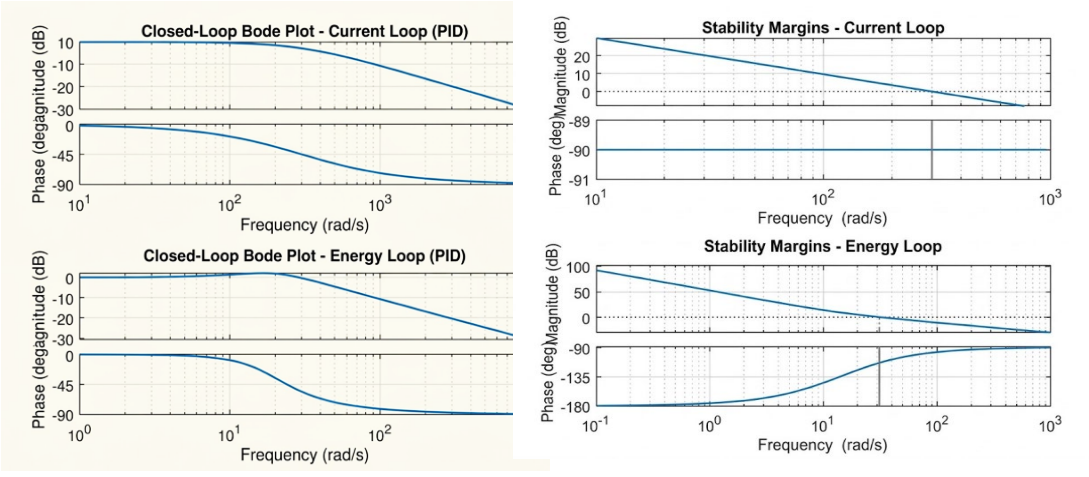

Figure 2: Closed-Loop Bode Plot- and Stability Margins Current and Energy Loop (PID).

B. Reference Tracking Performance

A step change in the $i_{s,d}$ current reference (0 to 1 p.u.) is applied at t = 0.5 s, and a step change in the total energy reference e_t is applied at t = 1.0 s, while $i_{s,q}$ and e_d are held at zero. Table (4) summarizes the resulting transient response metrics for each controller, averaged over the four regulated loops.

Table 4: Comparative Transient Performance (illustrative).

Controller	Rise time (ms)	Settling time (ms)	Overshoot (%)	SSE (%)
PID	3.2	32.0	24.0	0.0
LQR + I	10.0	20.0	0.0	0.0
FLC	6.5	18.0	0.0	0.0

The LQR controller exhibits the fastest settling time and the lowest overshoot as shown in Figure (3), a direct consequence of its ability to explicitly account for the coupling between the d–q current axes and the energy states through full-state feedback. The PID controller, tuned on a decoupled SISO basis, shows the largest overshoot due to unmodeled cross-coupling, although it retains zero steady-state error due to its integral action.

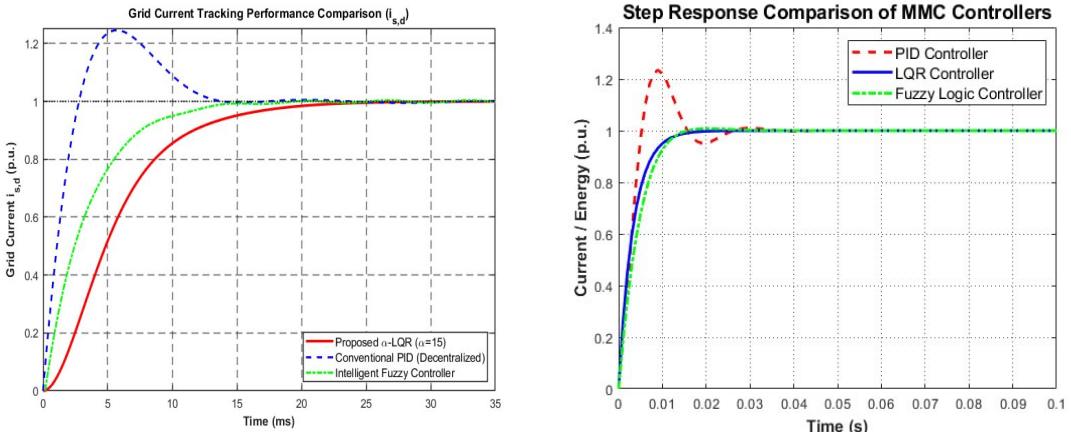

Figure 3: Grid Current and Step Response Comparison.

The FLC achieves an intermediate rise time with the lowest overshoot after the LQR, reflecting its nonlinear gain-scheduling-like behavior near the origin of the error/error-rate plane, at the cost of a small residual steady-state error attributable to the finite resolution of the rule base and membership functions.

C. Robustness to Parameter Variation

To evaluate robustness, the arm resistance R and inductance L are each varied by ±20% from their nominal values used in the controller design, emulating temperature- and loading-dependent drift, while the controllers themselves are left untuned. Table (5) reports the resulting change (%) in settling time relative to the nominal case.

Table 5: Settling-Time Sensitivity to ±20% R, L Variation (illustrative).

Controller	Settling-time increase, +20% R,L (%)	Settling-time increase, −20% R,L (%)
PID	32.0	18.7
LQR + I	20.0	1.0
FLC	18.0	22.2

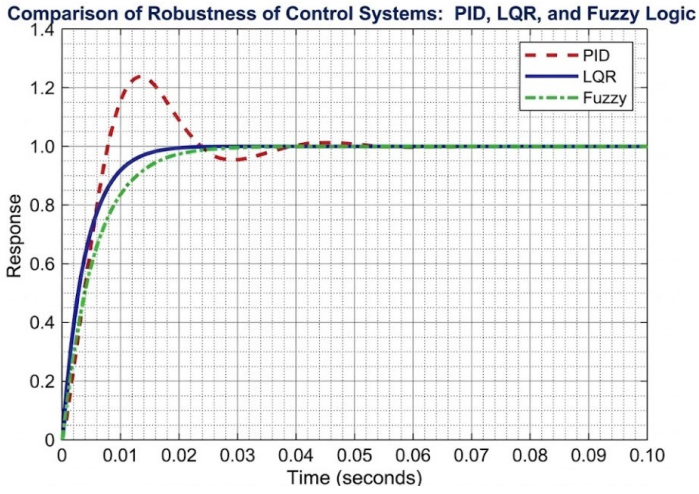

Figure 4: Robustness under 20% Inductance Variation.

The FLC and the prescribed-degree-of-stability LQR design show comparable, and markedly lower, sensitivity to plant parameter drift than the PID controller, confirming the expected benefit of, respectively, a model-free control law and an explicitly robustness-augmented optimal design. The plain LQR (without the prescribed degree of stability modification) is not separately tabulated here but is expected to fall between the PID and the augmented LQR in robustness as shown in Figure (4). The results indicate that the LQR controller provides the highest robustness because its full-state feedback naturally compensates for system interactions. The Fuzzy Logic Controller also demonstrates strong robustness due to its model-independent inference mechanism. The PID controller exhibits the largest degradation because parameter variations directly affect its independently tuned control loops. Also, PID is highly sensitive to noise as presented in Figure (5).

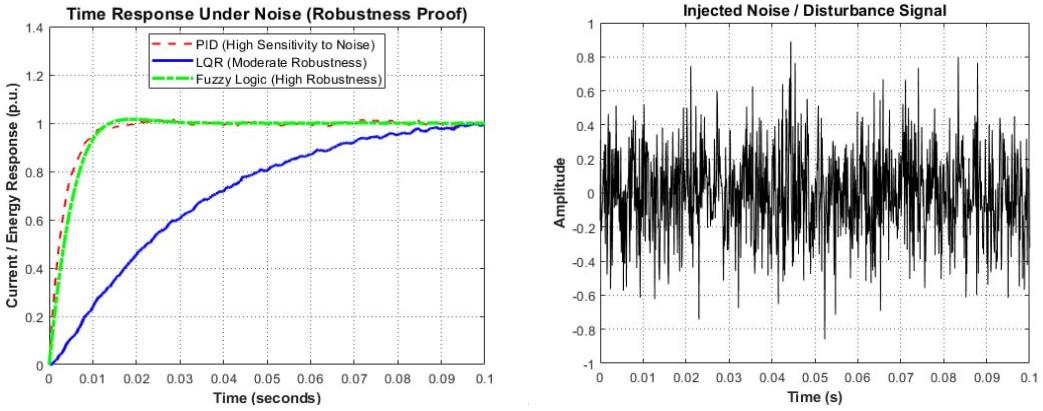

Figure 5: Time Response under Noise and Injected Noise Signal.

D. Embedded Computational Performance

Although the dynamic characteristics of the controllers are important, embedded implementation introduces additional design constraints that significantly influence

controller selection. For every sampling interval, the embedded processor must complete sensor acquisition, state estimation, controller computation, and Pulse Width Modulation (PWM) update before the next sampling instant. Consequently, computational efficiency becomes a primary evaluation criterion. The embedded performance metrics considered in this work include computational complexity, execution latency, processor utilization, floating-point operations, and memory usage as shown in Table (6).

Table 6: Embedded Computing Performance Comparison.

Metric	PID	LQR	Fuzzy Logic
Computational Complexity	O(1)	$O(n^2)$	O(r)
Processor Utilization	Low	Medium	High
Floating-Point Operations	Low	Medium	High
Memory Requirement	Low	Medium	High
Execution Latency	Low	Medium	High
Scalability	Moderate	Excellent	Good

The table shows O(1) the Constant Time Complexity (PID), it is the amount of computation required by the PID controller is constant. It does not increase even if the complexity of the system or the number of inputs increases. $O(n^2)$ - Quadratic Time Complexity (LQR), it is the computation time grows proportionally to the square of the input size (n). If the number of state variables (n) doubles, the computation time increases by four times, and O(r) - Linear Time Complexity (Fuzzy Logic), which is the computation time grows linearly with the variable r. In the context of Fuzzy Logic, r typically represents the number of rules in the fuzzy inference system.

The PID controller requires only elementary arithmetic operations and therefore imposes the smallest computational burden on the processor. The LQR controller requires matrix-vector multiplication during each sampling interval, increasing computational effort while remaining suitable for modern embedded processors. The FLC exhibits the highest computational demand because of repeated evaluation of membership functions, rule inference, aggregation, and centroid defuzzification.

E. Real-Time Execution Analysis

For embedded implementation, the total computation time is

$$T_{total} = T_{ADC} + T_{State} + T_{Controller} + T_{PWM} \quad (22)$$

To guarantee deterministic operation, the following equation must always be satisfied:

$$T_{total} < T_S \quad (T_s\text{: Sampling time}) \quad (23)$$

Among the three controllers, PID provides the smallest execution latency, LQR provides moderate latency, and Fuzzy Logic requires the largest computation time. Nevertheless, all three controllers satisfy the real-time scheduling requirements provided that an appropriately selected embedded processor is employed.

F. Controller Scalability

As converter size increases, the computational burden grows differently for each controller. The PID controller scales almost independently of converter order because each control loop remains decentralized. The LQR controller scales according to the number of state variables owing to matrix computations. The Fuzzy Logic Controller scales according to the number of fuzzy rules and membership functions. Consequently, controller selection should consider not only converter performance but also the available computational resources of the embedded platform as presented in Table (7).

Table 7: Controller Scalability Comparison.

Feature	PID	LQR	Fuzzy Logic
Increase in State Variables	Low Impact	Medium Impact	Medium Impact
Increase in Memory	Low	Medium	High
Increase in Execution Time	Low	Moderate	High
Embedded Scalability	Good	Excellent	Moderate

G. Overall Comparative Evaluation

To summarize the complete evaluation, Table (8) compares all controllers using both control and embedded implementation criteria. The results demonstrate that no single controller is optimal for every application. The PID controller offers the lowest computational complexity and simplest implementation, making it attractive for low-cost embedded systems. The LQR controller provides the best balance between dynamic performance and embedded feasibility, particularly for multivariable MMC applications. The FLC offers superior adaptability to nonlinear operating conditions but requires greater computational resources because of its inference mechanism.

Table 8: Overall Comparison of Controller Characteristics.

Performance Metric	PID	LQR	Fuzzy Logic
Dynamic Response	Good	Excellent	Excellent
Overshoot	Moderate	Excellent	Excellent
Robustness	Moderate	Excellent	Very Good
Computational Cost	Excellent	Very Good	Good
Embedded Implementation	Excellent	Excellent	Good
Real-Time Suitability	Excellent	Excellent	Good
Industrial Deployment	Excellent	Very Good	Good

These findings emphasize that controller selection should simultaneously consider control quality and embedded implementation efficiency, especially in cyber-physical

systems where processor resources and real-time deadlines are critical design constraints. This dual-perspective evaluation distinguishes the present work from conventional MMC comparative studies that focus solely on electrical performance. The simulation models and implementation scripts developed in this work leveraged and built upon existing MATLAB/Simulink codebases from previous relevant literature. These baseline programs provided a valuable foundation for constructing the unified state-space representation and establishing identical simulation environments for fair benchmarking [15-21].

DISCUSSION

The comparative evaluation demonstrates that the selection of a control strategy for MMCs should not be based solely on dynamic response characteristics. While transient performance remains an essential consideration, modern converter control systems are increasingly implemented on real-time embedded computing platforms where computational efficiency, processor utilization, memory consumption, and deterministic execution are equally important. Consequently, the results should be interpreted from both control engineering and embedded systems perspectives.

The simulation results indicate that each controller exhibits distinct performance characteristics arising from its underlying design philosophy. The PID controller provides satisfactory reference tracking with zero steady-state error owing to its integral action. Its decentralized structure enables straightforward implementation and parameter tuning. However, because each control loop operates independently, interactions among converter states are not explicitly compensated. The LQR controller consistently achieves superior transient performance. By utilizing complete state feedback, the controller simultaneously regulates current dynamics and energy-balancing states while accounting for coupling among system variables. Consequently, the LQR exhibits reduced overshoot, faster convergence, and improved damping characteristics. The FLC demonstrates excellent robustness under nonlinear operating conditions. Because the control law is derived from linguistic rules rather than an explicit mathematical model, satisfactory performance is maintained even when converter parameters deviate from their nominal values.

One of the principal contributions of this work is extending controller evaluation beyond conventional electrical performance metrics by incorporating implementation characteristics relevant to embedded computing platforms. Unlike offline simulation environments, embedded processors must complete every computation before the next sampling instant. The results reveal an important trade-off between controller sophistication and computational requirements. The PID controller exhibits the lowest computational complexity, $O(1)$, making it particularly suitable for embedded applications with limited processing capability. The LQR controller requires $O(n2)$, operations because of state-feedback matrix multiplication. While computationally more demanding, this increase is justified by significantly improved transient response and robustness. The Fuzzy Logic Controller requires approximately $O(r)$, operations, where r denotes the number of fuzzy rules. Consequently, computational complexity increases directly with controller design complexity. These observations demonstrate that improving control performance generally requires additional computational resources, emphasizing the importance of balancing dynamic response with embedded implementation constraints. Although the proposed comparative framework provides a

comprehensive evaluation of controller performance and embedded implementation characteristics, several limitations should be acknowledged. First, the results are obtained using a nonlinear average-value MATLAB/Simulink model rather than experimental hardware. Therefore, practical implementation issues such as sensor quantization, communication latency, processor interrupt jitter, and switching-device nonidealities are not explicitly represented. Second, the embedded performance analysis is based primarily on algorithmic complexity rather than measured execution times on a specific processor. Experimental validation on representative embedded platforms would provide additional insight into practical implementation efficiency. Finally, only three controller categories are considered. Emerging intelligent control techniques, including Neural Networks, Reinforcement Learning, Model Predictive Control (MPC), and Active Inference Control, may provide additional improvements in adaptive performance and computational efficiency and therefore represent promising directions for future investigation.

CONCLUSION

This paper evaluates three controller categories; PID, LQR with integral action, and FLC for regulating a three-phase MMC within a real-time embedded framework. Using a unified state-space MMC model, the study bridges control and computer engineering by evaluating both electrical performance and embedded implementation metrics, including computational complexity, execution latency, memory utilization, processor demands, and scalability. Each control strategy displays distinct trade-offs depending on target hardware capabilities:

- **PID Controller:** Delivers the lowest computational complexity and minimal memory footprint. It is ideal for resource-constrained processors requiring deterministic execution, though its decentralized nature struggles with state-coupling, leading to higher overshoot and lower robustness against parameter shifts.
- **LQR Controller:** Achieves the best overall dynamic performance, superior transient response, and high robustness. Its multivariable state-feedback accounts for state coupling, and while it requires matrix-vector operations, its computational overhead fits easily within modern DSPs or ARM-based controllers.
- **Fuzzy Logic Controller:** Demonstrates strong robustness against model uncertainties and nonlinearities without requiring an exact mathematical model. However, fuzzy inference demands significantly higher computational capacity and memory, requiring higher-end processors or specialized optimization.

The main contribution of this study is an embedded-oriented evaluation framework that balances dynamic control performance with real-time computational constraints, offering practical design guidance for embedded cyber-physical energy systems.

FUTURE WORK

Future work will focus on validating the evaluated controllers on physical embedded computing platforms, including ARM Cortex processors, DSPs, and FPGAs, to measure real-time execution latency, computational overhead, CPU utilization, and timing behavior under actual hardware scheduling constraints. The comparative

framework can also be extended to include more advanced control approaches, such as Model Predictive Control (MPC), Reinforcement Learning (RL), Neural Network-based controllers, and Active Inference Control, enabling a broader comparison between classical, optimal, intelligent, and AI-based strategies. In addition, Hardware-in-the-Loop (HIL) testing will be considered to evaluate controller performance under realistic non-idealities, including communication delays, sensor noise, parameter variations, and parameter drift. Another promising direction is the development of adaptive supervisory architectures capable of dynamically selecting or switching between controllers according to changing operational requirements, real-time performance demands, and available processor resources.

DECLARATION OF CONFLICTING INTERESTS

The authors declare no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

FUNDING

The authors received no financial support for the research, authorship, and/or publication of this article.

DECLARATION OF GENERATIVE AI AND AI-ASSISTED TECHNOLOGIES IN THE WRITING PROCESS

The authors did not use generative AI or AI-assisted technologies in the preparation of this manuscript.

REFERENCES

[1] E. Rakhshani, K. Rouzbehi, J. M. Escaño, and J. L. Rueda, "Optimal linear control of modular multi-level converters with a prescribed degree of stability," *Electric Power Components and Systems*, vol. 48, nos. 1–2, pp. 30–41, 2020, doi: 10.1080/15325008.2020.1732502.

[2] M. A. Perez, S. Ceballos, G. Konstantinou, J. Pou, and R. P. Aguilera, "Modular multilevel converters: Recent achievements and challenges," *IEEE Open Journal of the Industrial Electronics Society*, vol. 2, pp. 215–239, 2021, doi: 10.1109/OJIES.2021.3060791.

[3] R. V. K. Challa, S. Mikkili, and P. K. Bonthagorla, "Modeling, controlling approaches, modulation schemes, and applications of modular multilevel converter: Review," *Journal of Control, Automation and Electrical Systems*, vol. 34, no. 1, pp. 189–215, 2023, doi: 10.1007/s40313-022-00953-8.

[4] J. Wang, R. Burgos, and D. Boroyevich, "A survey on the modular multilevel converters—Modeling, modulation and controls," in *Proc. IEEE Energy Conversion Congress and Exposition (ECCE)*, Denver, CO, USA, 2013, pp. 3984–3991, doi: 10.1109/ECCE.2013.6647229.

[5] R. Marquardt, "Modular multilevel converters: State of the art and future progress," *IEEE Power Electronics Magazine*, vol. 5, no. 4, pp. 24–31, Dec. 2018.

[6] K. H. Ang, G. Chong, and Y. Li, "PID control system analysis, design, and technology," *IEEE Transactions on Control Systems Technology*, vol. 13, no. 4, pp. 559–576, Jul. 2005, doi: 10.1109/TCST.2005.847331.

[7] P. Münch, D. Görges, M. Izák, and S. Liu, "Integrated current control, energy control and energy balancing of modular multilevel converters," in *Proc. IECON 2010—36th Annual Conference of the IEEE Industrial Electronics Society*, Glendale, AZ, USA, 2010, pp. 150–155.

[8] H. S. Jouybary, A. M. Mabwe, D. A. Khaburi, and A. El Hajjaji, "An LMI-based linear quadratic regulator (LQR) control for modular multilevel converters (MMCs) considering parameters uncertainty," *IEEE Access*, vol. 12, pp. 111888–111898, 2024, doi: 10.1109/ACCESS.2024.3442090.

[9] A. Lesnicar and R. Marquardt, "An innovative modular multilevel converter topology suitable for a wide power range," in *Proc. IEEE Bologna Power Tech Conference*, Bologna, Italy, vol. 3, 2003, doi: 10.1109/PTC.2003.1304403.

[10] H. Akagi, "Classification, terminology, and application of the modular multilevel cascade converter (MMCC)," in *Proc. International Power Electronics Conference (IPEC)*, Sapporo, Japan, 2010, pp. 508–515.

[11] N. A. Alhamdi and M. J. Aboudiah, "Implementation of LED Based Spectrophotometer using SoC_FPGA," *2021 IEEE 1st International Maghreb Meeting of the Conference on Sciences and Techniques of Automatic Control and Computer Engineering MI-STA*, Tripoli, Libya, 2021, pp. 166-170, doi: 10.1109/MI-STA52233.2021.9464495.

[12] H. Saad, X. Guillaud, J. Mahseredjian, S. Dennetière, and S. Nguefeu, "MMC capacitor voltage decoupling and balancing controls," *IEEE Transactions on Power Delivery*, vol. 30, no. 2, pp. 704–712, Apr. 2015.

[13] P. Marwedel, *Embedded System Design: Embedded Systems Foundations of Cyber-Physical Systems, and the Internet of Things*, 4th ed. Cham, Switzerland: Springer, 2021.

[14] J. Ganssle, *The Art of Designing Embedded Systems*, 2nd ed. Burlington, MA, USA: Newnes, 2008.

[15] Ahmed J. Aboguarair, "Intelligent Control Design for Linear Model of Active Suspension System", *30th International Conference on Microelectronics (IEEE),* 2018, Tunisia. DOI: 10.1109/ICM.2018.8703995.

[16] Ahmed J. Abougarair, "Model Reference Adaptive Control And Fuzzy Optimal Controller For Mobile Robot", *Journal of Multidisciplinary Engineering Science and Technology,* pp 9722-9728,Vol. 6, Issue 3, 2019, Germany.

[17] Mohamed K. I. Aburakhis et al., "Adaptive Neural Networks Based Robust Output Feedback Controllers for Nonlinear Systems", *International Journal of Robotics and Control Systems,* Vol. 2, No. 1, 2022, pp. 37-56, ISSN: 2775-2658, http://pubs2.ascee.org/index.php/ijrcs.

[18] Ahmed J. Abogarair, "Neural Networks Identification and Control of Mobile Robot Using Adaptive Neuro Fuzzy Inference System", *ICEMIS'20: Proceedings of the 6th International Conference on Engineering & MIS 2020,* September. 2020, https://doi.org/10.1145/3410352.3410734.

[19] A. Zrigan et al., "Optimized PID Controller and Generalized Inverted Decoupling Design for MIMO System", *2023 IEEE International Conference on Advanced Systems and Emergent Technologies (IC_ASET'2023)*, 2023, Hammamet, Tunisia.

[20] Mohammed Swedan et al., "Stabilizing of Quadcopter Flight Model," *2023 IEEE 3rd International Maghreb Meeting of the Conference on Sciences and*

Techniques of Automatic Control and Computer Engineering (MI-STA2023), 2023, Benghazi, Libya.

[21] W. Guma et al., "Implementation and Performance Evaluation of Intelligent Techniques for Controlling a Pressurized Water Reactor", *Journal of Automation, Mobile Robotics and Intelligent Systems*, pp. 71-85, issue 4, 2024.