MITOR: Jurnal Teknik Elektro

Simulation and Comparative Analysis of PID and LQR Controllers for Speed Control of a BLDC Motor Based on State-Space Model

Faisal Wahab*
Departement of Electrical Engineering, Faculty of Engineering Technology, – Parahyangan Catholic University
Bandung, Indonesia
*faisal.wahab@unpar.ac.id

Abstract – *This study explores the performance of Proportional-Integral-Derivative (PID) and Linear Quadratic Regulator (LQR) controllers in regulating BLDC motor speed. The PID controller is tuned using both MATLAB-based auto-tuning and manual parameter selection, while the LQR controller is designed by adjusting the Q and R matrices through iterative optimization. To evaluate their effectiveness, a mathematical model of the BLDC motor is developed in a state-space representation, enabling simulation of its dynamic response. The open-loop system shows a slow response and significant steady-state error, making it unsuitable for precise speed control. The PID tuner improves response time and accuracy, it introduces overshoot. Manual tuning further refines performance by reducing overshoot and enhancing stability. The LQR controller demonstrates the fastest response, minimal overshoot, and the lowest steady-state error, making it the most effective approach for BLDC motor speed regulation. This research addresses a practical need for comparative controller performance in BLDC motor systems, which is essential for high-precision applications. The findings provide engineers with performance-based insights into choosing between PID and LQR under consistent simulation conditions.*

Keywords – *BLDC motor; state space; PID control; LQR control; simulation.*

I. Introduction

A Brushless Direct Current (BLDC) motor is an advanced type of electric motor that eliminates the need for mechanical brushes and commutators by employing electronic commutation. This innovation has led to the widespread adoption of BLDC motors in various fields such as robotics [1], electric vehicles [2], agriculture [3], and industrial automation [4]. Their widespread application is due to exceptional efficiency, high power density, and superior reliability [5], making them a preferred choice for modern applications requiring precise and robust motor performance.

However, achieving precise control of BLDC motors remains a significant challenge, particularly in terms of response time. To address these issues, various advanced control strategies have been developed to increase the performance of BLDC motors. Several controllers have been implemented for BLDC motors, including Proportional-Integral-Derivative (PID) [6], Fuzzy Logic [7], Fuzzy-PID [8], Linear Quadratic Regulator (LQR) [9], and others [10]. Each of these control methods offers unique advantages and compromises, contributing to improve motor performance and responsiveness.

Among these, Proportional-Integral-Derivative (PID) controllers are one of the most commonly used control methods, primarily due to their simplicity, ease of implementation, and effectiveness in a variety of applications [11]. PID controllers offer a robust solution for maintaining steady-state error at minimal levels while ensuring quick response to dynamic changes. However, their effectiveness may decline in highly nonlinear systems with complex system dynamics, often requiring manual tuning to achieve optimal performance.

In contrast, the Linear Quadratic Regulator (LQR) is an optimal control strategy that provides a more systematic approach to control by minimizing a predefined cost function [12]. The LQR controller is designed to balance state deviations and control effort, leading to improved performance in systems with multiple inputs and outputs [13]. It offers advantages such as faster convergence, superior stability, and smoother control compared to PID, especially in multi-variable systems [14]. However, LQR implementation is more

The manuscript was received on August 25, 2025, revised on June 14, 2025, and published online on July 25, 2025. `Emitor` is a Journal of Electrical Engineering at Universitas Muhammadiyah Surakarta with ISSN (Print) 1411 – 8890 and ISSN (Online) 2541 – 4518, holding Sinta 3 accreditation. It is accessible at https://journals2.ums.ac.id/index.php/emitor/index.

complex, requiring an accurate mathematical model of the system, including its state-space representation, and the solution of Riccati equations. This makes LQR more complex to compute and difficult to tune in practice. Linear Quadratic Regulator (LQR) is an optimal control method that requires solving Riccati equations and accurate system modeling [15, 16]. While effective for various applications like vehicle suspension control and rocket thrust vectoring [15, 17], LQR implementation can be complex, especially for high-order systems [16]. Researchers have proposed alternative approaches to address these challenges, including data-driven tuning methods [16], parallel implementations of Riccati recursion [18], and sensitivity-based tuning [19]. MATLAB has been utilized to solve LQR problems and associated Riccati equations [20]. The LQR method's effectiveness has been demonstrated in self-tuning control applications, offering improved performance over other control techniques [21]. Despite its complexity, LQR remains a powerful tool for optimal control in various engineering domains. Additionally, selecting optimal weighting matrices is critical for LQR performance, which can be challenging. Despite its limits with complex systems, PID remains a practical and widely used solution in many real-world applications due to its simplicity and lower modeling requirements. Meanwhile, LQR is more complex but provides better precision and stability. The choice between these controllers depends on the specific requirements and constraints of the application.

Therefore, this research evaluates the performance of PID and LQR controllers on a BLDC motor model by comparing their speed control capabilities and response times. Through simulation and analysis, the study aims to identify the relative advantages and compromises of these control methods. The study compares PID and LQR controllers using key metrics like rise time, settling time, and overshoot to evaluate their effectiveness in controlling a BLDC motor.

While many researchers have addressed PID and LQR applications independently, this paper aims to fill a research gap by directly comparing both methods under identical modeling conditions and metrics.

This study is significant in guiding engineers and researchers in selecting appropriate controllers based on performance metrics and system behavior in practical applications. It adds value by providing simulation evidence and performance analysis.

The structure of this paper is organized as follows: Section II presents the modeling of the BLDC motor and the control strategies applied. Section III explains the simulation procedures and compares the results. Section IV provides concluding remarks and suggestions for future research directions.

II. Research Methods

This section presents the methodology used to analyze and compare the performance of PID and LQR controllers for BLDC motor speed control. The approach includes mathematical modeling of the BLDC motor in state-space form, followed by the design and implementation of the two control strategies. Each step is detailed in the following subsections.

i. Modelling BLDC Motor

To model a Brushless Direct Current (BLDC) motor using the state-space representation, the system must be defined in terms of its state variables, and the differential equations governing its dynamics must be derived. The dynamics of a BLDC motor are typically described by electrical and mechanical equations involving the phase currents (i_a, i_b, i_c) and rotor speed (ω) and rotor position (θ). A BLDC motor is commonly modelled using three-phase stator windings [22]. Figure 1 illustrates the schematic circuits of a BLDC motor. The voltage equations for the three phases can be expressed in Equation 1.

$$\begin{aligned} V_a &= L_a \frac{di_a}{dt} + R_a i_a + e_a \\ V_b &= L_b \frac{di_b}{dt} + R_b i_b + e_b \\ V_c &= L_c \frac{di_c}{dt} + R_c i_c + e_c \end{aligned} \tag{1}$$

Where V_a, V_b, V_c are the phase voltages, i_a, i_b, i_c are the phase currents, L is the stator inductance, R is the stator resistance, and e_a, e_b, e_c are the back electromotive forces (EMF) generated in each phase.

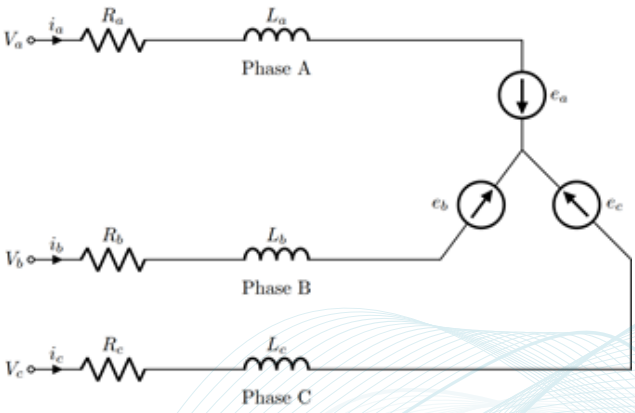

Figure 1: Schematic circuit of BLDC motor

The back EMFs e_a, e_b, e_c are functions of rotor speed (ω) and position (θ). For a BLDC motor,

these back EMFs are typically sinusoidal and can be expressed in Equation 2.

$$\begin{aligned} e_a &= K_e \omega f(\theta) \\ e_b &= K_e \omega f\left(\theta - \frac{2\pi}{3}\right) \\ e_c &= K_e \omega f\left(\theta + \frac{2\pi}{3}\right) \end{aligned} \tag{2}$$

Where K_e is the back-EMF constant, ω is the rotor angular speed, and $f(\theta)$ is the function representing the back-EMF waveform based on rotor position. These sinusoidal back EMF waveforms are a key characteristic of BLDC motors and play a critical role in determining the motor's electrical and mechanical behaviour. By using these equations into the state-space model, the BLDC motor's behaviour can be accurately described, allowing for better control design and analysis. The mechanical dynamics of the rotor can be modelled using Newton's second law of motion, as shown in Equation 3.

$$J\frac{d\omega}{dt} + B\omega = T_e - T_L \tag{3}$$

Where J is the moment of inertia of the rotor, B is the damping coefficient, T_e is the electromagnetic torque generated by the motor, T_L is the load torque, and ω is the rotor angular speed. The electromagnetic torque T_e is related to the currents and back EMFs as shown in Equation 4.

$$\begin{aligned} T_e = K_t \left[i_a f(\theta) \right] + i_b f\left(\theta - \frac{2\pi}{3}\right) \\ + i_c f\left(\theta + \frac{2\pi}{3}\right) \end{aligned} \tag{4}$$

Where K_t represents the torque constant. The rotor position θ is related to the rotor angular speed ω by Equation 5.

$$\frac{d\theta}{dt} = \omega \tag{5}$$

To model the BLDC motor in state-space form, the state variables are chosen as the phase currents i_a, i_b, i_c, the rotor angular velocity ω, and the rotor position θ. Thus, the state vector x is given in Equation 6.

$$x = \begin{bmatrix} i_a \\ i_b \\ i_c \\ \omega \\ \theta \end{bmatrix} \tag{6}$$

The state-space form can be expressed as Equation 7.

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \tag{7}$$

Where $\dot{x}$ is the time derivative of the state vector, A is the system matrix representing the dynamics, B is the input matrix, u is the input vector (phase voltages V_a, V_b, V_c), C is the output matrix, D is the feedthrough matrix, and y is the output vector.

State Equation for currents (i_a, i_b, i_c) for each phase, the current equation can be written as Equation 8.

$$\begin{aligned} \frac{di_a}{dt} &= \frac{1}{L}(V_a - R_a i_a - e_a) \\ \frac{di_b}{dt} &= \frac{1}{L}(V_b - R_b i_b - e_b) \\ \frac{di_c}{dt} &= \frac{1}{L}(V_c - R_c i_c - e_c) \end{aligned} \tag{8}$$

The mechanical equation can be rewritten as shown in Equation 9.

$$\frac{d\omega}{dt} = \frac{1}{J}(T_e - B\omega - T_L) \tag{9}$$

The equation for rotor position is given in Equation 10.

$$\frac{d\theta}{dt} = \omega \tag{10}$$

The matrices A, B, C and D describe how the state variables change with time, as shown in Equation 11.

$$\begin{aligned} A &= \begin{bmatrix} -\frac{R}{L} & 0 & 0 & -\frac{K_e}{L} & 0 \\ 0 & -\frac{R}{L} & 0 & -\frac{K_e}{L} & 0 \\ 0 & 0 & -\frac{R}{L} & -\frac{K_e}{L} & 0 \\ K_t & K_t & K_t & -\frac{B}{J} & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}, B = \begin{bmatrix} \frac{1}{L} & 0 & 0 \\ 0 & \frac{1}{L} & 0 \\ 0 & 0 & \frac{1}{L} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\ C &= \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \end{bmatrix}, D = 0 \end{aligned} \tag{11}$$

ii. PID Controller

Proportional-Integral-Derivative (PID) control is one of the most widely used feedback control strategies in engineering applications. It provides a simple way to control dynamic systems, such as motors and robotics by adjusting the control input based on the error between a desired setpoint and the actual output. The PID controller constantly changes the input to reduce the error, which is the difference between the setpoint and the actual output. A PID controller has three main components: Proportional (P), Integral (I), and Derivative (D). Each component responds to the error signal in a different way, contributing to the overall control effort. Figure 2 shows the block diagram of a PID Controller.

The proportional term produces an output that is proportional to the error value. It adjusts the control signal based on how far the process variable is from the setpoint, as shown in Equation 12.

$$u(t) = K_p e(t) \tag{12}$$

Where $u(t)$ denotes the control input, and K_p represents the proportional gain. In the diagram, the error

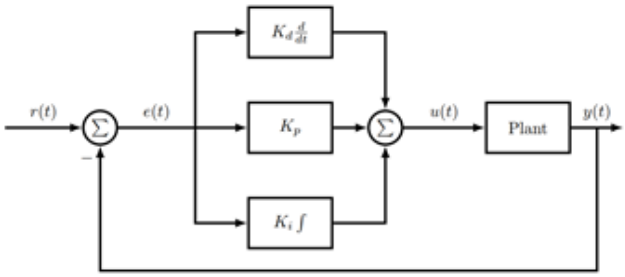

Figure 2: Diagram block of PID controller

is defined as $e(t) = r(t) - y(t)$, where $r(t)$ is the setpoint and $y(t)$ is the measured output. The larger the error, the stronger the control action.

The integral term eliminates steady-state error by integrating the error over time. It sums up past errors to ensure that the accumulated error is driven to zero, as shown in Equation 13.

$$u(t) = K_i \int e(\tau)\, d\tau \tag{13}$$

Where K_i denotes the integral gain. The integral increases the control input based on the accumulated error, helping the system to reach and stay at the desired setpoint. However, excessive integral action can cause oscillations and instability.

The derivative term predicts future error by considering the rate of change of the error. It responds to the slope of the error curve, helping to dampen oscillations and improve system stability, as shown in Equation 14.

$$u(t) = K_d \frac{de(t)}{dt} \tag{14}$$

Where K_d denotes the derivative gain. The overall PID control law is a combination of the three components, as shown in Equation 15.

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

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

iii. LQR Controller

The second controller is LQR (Linear Quadratic Regulator). LQR control is an optimal control method used in state-space representation to minimize a quadratic cost function. It is commonly used in control systems to maintain both performance and energy efficiency. The goal of the LQR is to find a control input $u(t)$ that drives the system's states $x(t)$ to the desired equilibrium point (usually $x = 0$) while minimizing the cost function. The cost function in LQR control defines the balance between reducing state deviations and minimizing control effort, as shown in Equation 16.

$$J = \int_0^\infty \left(x^T(t) Q x(t) + u^T(t) R u(t) \right) dt \tag{16}$$

Where J is the total cost to be minimized, Q is a positive semi-definite matrix that penalizes deviations from the desired state, $x(t)$ is the state vector of the system, R is a positive definite matrix that penalizes the control effort $u(t)$, and $x^T(t)$ and $u^T(t)$ are the transpose of the state and control input vectors. Increasing Q would improve tracking accuracy but may increase control input magnitude, while increasing R reduces control input magnitude but may lead to a slower response.

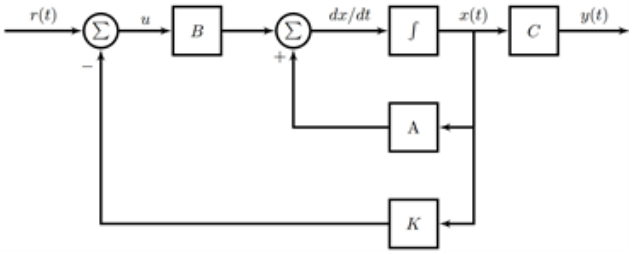

Figure 3: Block diagram of LQR controller

Figure 3 shows the block diagram of the LQR controller. Based on this diagram, the state-space equation can be rewritten as shown in Equation 17:

$$\begin{aligned} \dot{x} &= (A - BK)x + Bu \\ y &= Cx \end{aligned} \tag{17}$$

The LQR finds the control input $u(t)$ that minimizes the cost function J. The optimal control law for the system is a state-feedback law, as shown in Equation 18:

$$u(t) = -Kx(t) \tag{18}$$

Where K is the optimal gain matrix and $x(t)$ is the state vector. The gain matrix K is calculated using Equation 19:

$$K = R^{-1} B^T P \tag{19}$$

The matrix P is the solution to the Algebraic Riccati Equation (ARE) as shown in Equation 20:

$$A^T P + PA - PBR^{-1}B^T P + Q = 0 \tag{20}$$

The matrix P solves this equation and defines the best balance between state deviations and control effort.

III. Results and Discussion

In this discussion, four simulations were conducted: open-loop control, closed-loop control with a manually

tuned PID, closed-loop control with an automatically tuned PID, and closed-loop control with LQR. Prior to the simulations, the parameters of the BLDC motor were identified to implement the state-space model. The designed motor parameters are presented in Table 1.

Table 1: Parameter of BLDC Motor

Parameter	Value	Description
L	0.001	Phase inductance (H)
R	0.5	Phase resistance (Ω)
K_t	0.1	Torque constant (Nm/A)
K_b	0.01	Back EMF constant (V·s/rad)
J	0.01	Moment of inertia (kg·m^2)
b	0.1	Damping coefficient (Nms)

i. Open Loop System

After configuring the parameters, the first simulation was conducted under open-loop conditions. The result, shown in Figure 4, presents the motor speed (rad/s) over time (s) in response to a step input.

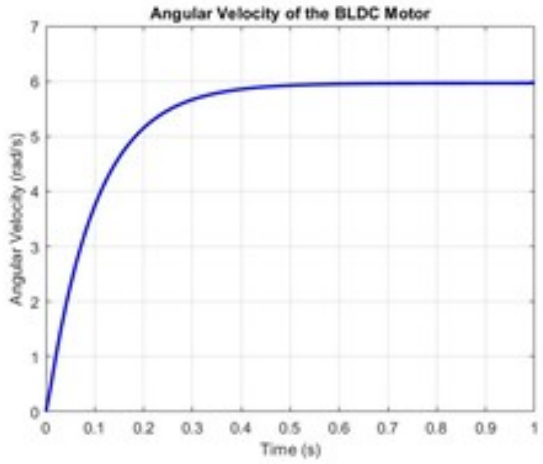

Figure 4: Simulation of BLDC motor in open-loop system

The performance metrics obtained from Figure 4 are as follows: rise time of 0.2183 s, settling time of 0.3909 s, peak time of 3.4100 s, and a steady-state error of 4.9642. Notably, the system demonstrates zero overshoot, indicating stability. However, the steady-state error suggests that controller implementation is necessary for improved accuracy and response.

ii. Speed Control using PID Controller

The second simulation introduces a PID controller to regulate the BLDC motor's speed. Two approaches are considered: automatic tuning via MATLAB's PID Tuner and manual tuning.

ii.1. Automatic PID Tuning

The setpoint remains at 6 rad/s, consistent with the open-loop response. From MATLAB's PID Tuner, the parameters obtained are $K_p = 0.085$, $K_i = 14.5892$, and $K_d = 0$. Figure 5 and Figure 6 present the system response and tracking error, respectively.

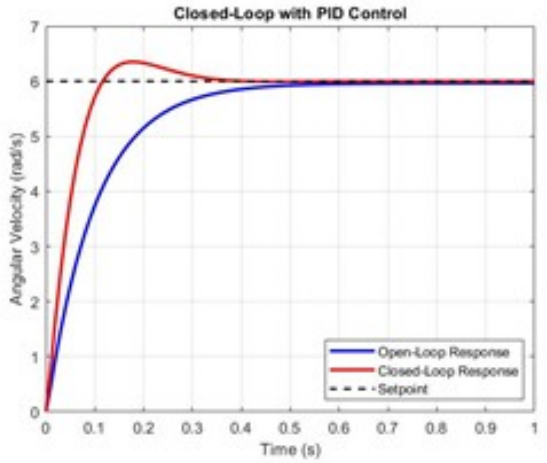

Figure 5: Implementation of PID control with MATLAB's PID tuner parameters

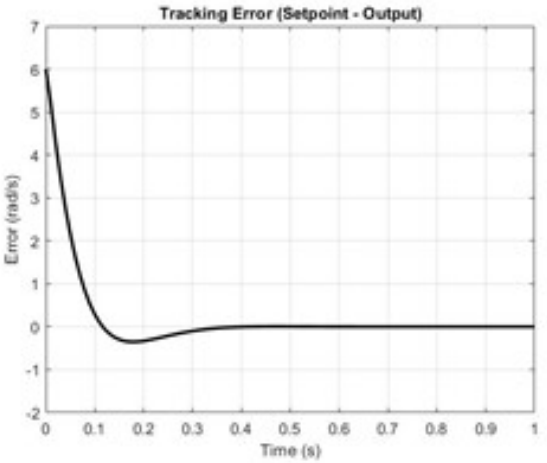

Figure 6: Tracking error in PID control (MATLAB PID Tuner)

Based on Figures 5 and 6, the PID-controlled system achieves a rise time of 0.0802 s, settling time of 0.2892 s, overshoot of 5.8616%, and peak time of 0.1800 s. While faster than the open-loop system, the overshoot suggests further optimization is needed.

ii.2. Manual PID Tuning

A second trial using manually tuned PID values—$K_p = 3$, $K_i = 30$, and $K_d = 0$—was conducted to improve performance. The resulting output and error tracking are shown in Figure 7 and Figure 8.

As depicted in Figures 7 and 8, the system

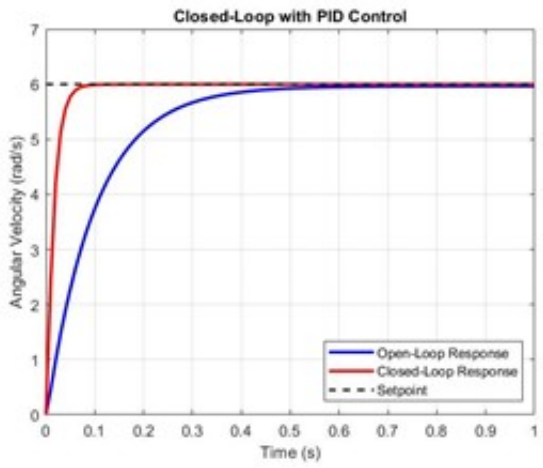

Figure 7: Implementation of PID control with manually tuned parameters

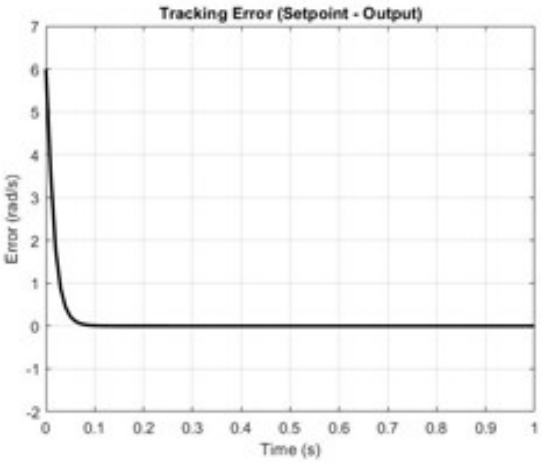

Figure 8: Tracking error in PID control (manual tuning)

achieved a rise time of 0.0340 s, settling time of 0.0594 s, overshoot of $2.2204 \times 10^{-14}\%$, and a peak time of 3.2700 s. These results confirm improved performance over the PID tuner method, with significantly reduced overshoot and enhanced responsiveness.

iii. Speed Control using LQR Controller

The fourth simulation utilizes Linear Quadratic Regulation (LQR) Control. Before implementing LQR, a controllability check is performed using the controllability matrix defined in Equation (21).

$$C = \begin{bmatrix} B & AB & A^2B & \cdots & A^{n-1}B \end{bmatrix} \tag{21}$$

Where n is the number of states and A and B are the system and input matrices, respectively. Using MATLAB's `ctrb` and `rank` functions, the result confirms that the system is controllable since the matrix has full rank (5), which equals the number of state variables.

The LQR controller utilizes the weighting matrices Q and R, defined in Equation (22).

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 0 & 0.1 \end{bmatrix}, R = \begin{bmatrix} 0.01 & 0 & 0 \\ 0 & 0.01 & 0 \\ 0 & 0 & 0.01 \end{bmatrix} \tag{22}$$

A desired state x_d (setpoint) can be incorporated into the cost function to reduce deviation [23], as modified in Equation (23):

$$J = \int_0^\infty \left[(x(t) - x_d)^T Q (x(t) - x_d) + u(t)^T R u(t) \right] dt \tag{23}$$

Where $x_d = \begin{bmatrix} 0 & 0 & 0 & \omega_d & 0 \end{bmatrix}$ represents the desired rotor speed. The LQR feedback law is applied in Equation (24).

$$u(t) = -K(x(t) - x_d) \tag{24}$$

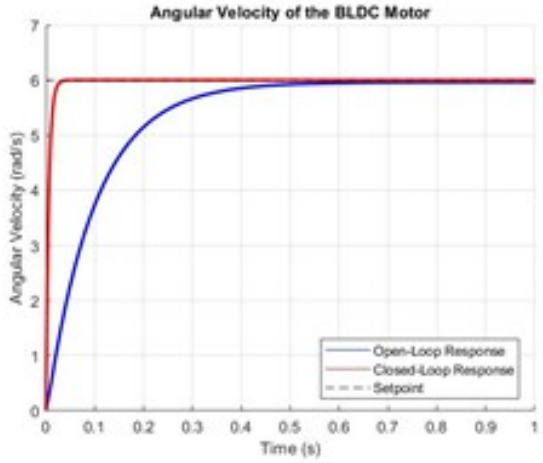

Figure 9: Implementation of LQR Control

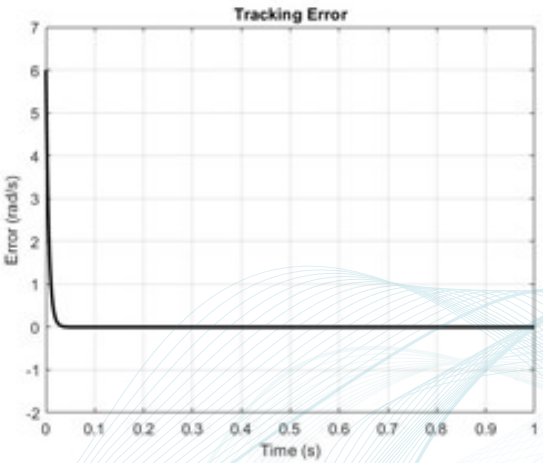

Figure 10: Tracking Error in LQR Control

Figures 9 and 10 show the speed response and tracking error of the BLDC motor with LQR control. The results indicate a rise time of 0.0127 s, settling time of 0.0226 s, overshoot of 0.0185%, and peak time of 0.0992 s. This demonstrates that LQR achieves superior transient performance compared to PID controllers.

iv. Comparison of Transient Response

Table 2: Comparison of Transient Response Specification

Response Time	Open Loop	PID Tuner	Tuned PID	LQR
Rise Time (s)	0.2183	0.0802	0.0340	0.0127
Overshoot (%)	0	5.8616	2.2204×10^{-14}	0.0185
Peak Time (s)	3.4600	0.1800	3.2700	0.0992
Settling Time (s)	0.3909	0.2892	0.0594	0.0226
Steady-State Error (%)	4.9642	3.5527×10^{-15}	1.7764×10^{-15}	8.1024×10^{-4}

Table 2 summarizes the performance of the BLDC motor under different control scenarios. The open-loop system shows the slowest response and largest error. PID tuning significantly improves the response, with the manually tuned PID showing better optimization. However, the LQR controller exhibits the most optimal performance, with the fastest rise time, lowest overshoot, and minimal steady-state error, confirming it as the most effective control strategy.

IV. Conclusion

The implementation of PID and LQR control strategies for a BLDC motor model was successfully conducted through simulation. Four test conditions were evaluated: open-loop, PID with MATLAB auto-tuning, PID with manual tuning, and LQR control. The evaluation focused on key transient metrics such as rise time, overshoot, peak time, and settling time. Results showed that the LQR controller outperformed all others by delivering the fastest rise time (0.0127 s), minimal overshoot (0.0185%), and shortest settling time (0.0226 s). Although PID controllers showed considerable improvements over the open-loop response—especially the manually tuned version—the LQR strategy proved most effective for precision and stability. Therefore, LQR is recommended for high-performance applications in BLDC motor control.

References

[1] S. Procter and E. Secco, "Design of a biomimetic bldc driven robotic arm for teleoperation & biomedical applications," *Journal of Human, Earth, and Future*, vol. 2, 2022. [Online]. Available: https://doi.org/10.28991/HEF-2021-02-04-03

[2] M. Azri *et al.*, "A review on bldc motor application in electric vehicle (ev) using battery, supercapacitor and hybrid energy storage system: Efficiency and future prospects," *Journal of Advanced Research in Applied Sciences and Engineering Technology*, vol. 30, pp. 41–59, 2023. [Online]. Available: https://doi.org/10.37934/araset.30.2.4159

[3] R. P. Antony, P. R. G. Komarasamy, N. Rajamanickam, R. Alroobaea, and Y. Aboelmagd, "Optimal rotor design and analysis of energy-efficient brushless dc motor-driven centrifugal monoset pump for agriculture applications," *Energies (Basel)*, vol. 17, no. 10, 2024. [Online]. Available: https://doi.org/10.3390/en17102280

[4] S. Sivaranjani and R. Ramachandran, "Internet of things based industrial automation using brushless dc motor application with resilient directed neural network control fed virtual z-source multilevel inverter topology," *Wirel Pers Commun*, vol. 102, 2018. [Online]. Available: https://doi.org/10.1007/s11277-018-5365-6

[5] G. Thenmozhi, S. Sivakumar, B. Arun, S. Sivakumar, M. Nirmala, and A. Radhika, "An investigation on the performance of permanent magnet brushless dc motor based on different materials," *AIP Conf Proc*, vol. 2446, no. 1, p. 100004, 2022. [Online]. Available: https://doi.org/10.1063/5.0108137

[6] S. Usha, P. Dubey, R. Ramya, and M. Suganyadevi, "Performance enhancement of bldc motor using pid controller," *International Journal of Power Electronics and Drive Systems (IJPEDS)*, vol. 12, p. 1335, 2021. [Online]. Available: https://doi.org/10.11591/ijpeds.v12.i3.pp1335-1344

[7] C. Hermanu and B. Apribowo, "Fuzzy logic controller and its application in brushless dc motor (bldc) in electric vehicle-a review," 2021, no DOI found.

[8] A. L. Shuraiji and S. W. Shneen, "Fuzzy logic control and pid controller for brushless permanent magnetic direct current motor: A comparative study," *Journal of Robotics and Control (JRC)*, vol. 3, no. 6, pp. 762–768, 2022. [Online]. Available: https://doi.org/10.18196/jrc.v3i6.15974

[9] S. Sharma *et al.*, "Optimal bldc motor control using a woa-based lqr strategy," 2022. [Online]. Available: https://doi.org/10.1109/GPECOM55404.2022.9815609

[10] I. Anshory, D. Hadidjaja, and I. Sulistiyowati, "Measurement, modeling, and optimization speed control of bldc motor using fuzzy-pso based algorithm," *Journal of Electrical Technology UMY (JET-UMY)*, vol. 5, no. 1, 2021, no DOI found.

[11] M. Mahmud, M. R. Islam, S. M. A. Motakabber, M. D. A. Satter, K. E. Afroz, and A. K. M. A. Habib, "Control speed of bldc motor using pid," in *2022 IEEE 18th International Colloquium on Signal Processing & Applications (CSPA)*, 2022, pp. 150–154. [Online]. Available: https://doi.org/10.1109/CSPA55076.2022.9782030

[12] T. Wang, H. Wang, H. Hu, and C. Wang, "Lqr optimized bp neural network pi controller for speed control of brushless dc motor," *Advances in Mechanical Engineering*, vol. 12, no. 10, 2020. [Online]. Available: https://doi.org/10.1177/1687814020968980

[13] P. Duan, L. He, Z. Duan, and L. Shi, "Linear quadratic regulator design for multi-input systems with a distributed cooperative strategy," 2021, available online. [Online]. Available: https://api.semanticscholar.org/CorpusID:243861214

[14] P. Saraf, M. Gupta, and A. M. Parimi, "A comparative study between a classical and optimal controller for a quadrotor," in *2020 IEEE 17th India Council International*

Conference (INDICON), 2020, pp. 1–6. [Online]. Available: https://doi.org/10.1109/INDICON49873.2020.9342485

[15] K. J. Wakeham and D. G. Rideout, "Model Complexity Requirements in Design of Half Car Active Suspension Controllers," in *ASME 2011 Dynamic Systems and Control Conference and Bath/ASME Symposium on Fluid Power and Motion Control, Volume 2*. ASMEDC, jan 1 2011. [Online]. Available: http://dx.doi.org/10.1115/DSCC2011-5955

[16] Z. Cheng, X. Li, J. Ma, C. S. Teo, K. K. Tan, and T. H. Lee, "Data-Driven Tuning Method for LQR Based Optimal PID Controller," in *IECON 2019 - 45th Annual Conference of the IEEE Industrial Electronics Society*. IEEE, 10 2019, pp. 5186–5191. [Online]. Available: http://dx.doi.org/10.1109/IECON.2019.8927075

[17] L. Sopegno, P. Livreri, M. Stefanovic, and K. P. Valavanis, "Linear Quadratic Regulator: A Simple Thrust Vector Control System for Rockets," in *2022 30th Mediterranean Conference on Control and Automation (MED)*. IEEE, jun 28 2022, pp. 591–597. [Online]. Available: http://dx.doi.org/10.1109/MED54222.2022.9837125

[18] F. G. and B. J. J., "Parallel Implementation of Riccati Recursion for Solving Linear-Quadratic Control Problems," 2013.

[19] D. Masti, M. Zanon, and A. Bemporad, "Tuning LQR Controllers: A Sensitivity-Based Approach," *IEEE Control Systems Letters*, vol. 6, pp. 932–937, 2022. [Online]. Available: http://dx.doi.org/10.1109/LCSYS.2021.3087556

[20] A. Abiodun, "The USE OF MATLAB IN THE SOLUTION OF LINEAR QUADRATIC REGULATOR (LQR) PROBLEMS," 2014.

[21] D. W. CLARKE, P. P. KANJILAL, and C. MOHTADI, "A generalized LQG approach to self-tuning control Part II. Implementation and simulation," *International Journal of Control*, vol. 41, no. 6, pp. 1525–1544, 6 1985. [Online]. Available: http://dx.doi.org/10.1080/0020718508961213

[22] T. M. Krishna, N. V. Gowri, G. S. Babu, B. Akshita, G. Agarwal, and M. Netha, "Modelling and development of controller for bldc motor," *International Journal of Electrical Engineering and Technology (IJEET)*, vol. 12, no. 6, pp. 200–210, 2021. [Online]. Available: https://doi.org/10.34218/IJEET.12.6.2021.019

[23] B. Ata and M. Gencal, "Comparison of optimization approaches on linear quadratic regulator design for trajectory tracking of a quadrotor," dec 20 2023. [Online]. Available: https://doi.org/10.21203/rs.3.rs-3762975/v1