IJRCS
International Journal of Robotics and Control Systems
Vol. 6, No. 1, 2026, pp. 1-15
ISSN 2775-2658
http://pubs2.ascee.org/index.php/ijrcs
ASCEE

Basic Tutorial on Model Predictive Control for Speed Control of DC-motor

Hari Maghfiroh [a,1,*], Joko Slamet Saputro [a,2], Agus Ramelan [a,3], Alfian Ma'arif [b,4], Isa Ali Ibrahim [c,5], Smail Latifa [d,6]

[a] Dept. of Electrical Engineering, Universitas Sebelas Maret, Surakarta, Indonesia
[b] Dept. of Electrical Engineering, Universitas Ahmad Dahlan, Yogyakarta, Indonesia
[c] School of Information and Communications Technology, Federal University of Technology Owerri, Imo State, Nigeria
[d] LSETER Laboratory, Dep. Electrical Engineering, Nour Bachir University Center, El Bayadh, Algeria
[1] hari.maghfiroh@staff.uns.ac.id; [2] jssaputro89@staff.uns.ac.id; [3] agusramelan@staff.uns.ac.id;
[4] alfian.maarif@te.uad.ac.id; [5] ibrahim.ali@futo.edu.ng; [6] l.smail@cu-elbayadh.dz
*Corresponding Author

ARTICLE INFO

Article history
Received June 04, 2025
Revised August 01, 2025
Accepted December 2, 2025

Keywords
MPC;
PID;
Optimal Control;
Speed Control;
DC Motor

ABSTRACT

This tutorial paper presents a foundational introduction to Model Predictive Control (MPC) using DC motor speed control as a practical case study. While Proportional-Integral-Derivative (PID) controllers remain widely used in industry due to their simplicity, they often lack the ability to manage system constraints or anticipate future behavior. In contrast, MPC offers a predictive, optimization-based approach that considers such constraints over a finite prediction horizon. To evaluate its effectiveness, MPC is compared with PID across three simulation-based tests: unit step response, input disturbance rejection, and signal tracking. The results show that, within the simulated environment, MPC achieves improved control performance, reducing settling time by 75%, overshoot by 76.6%, and Integral of Absolute Error (IAE) by 49% in step response tests. Under input disturbances and dynamic set-point changes, MPC also achieves lower IAE and faster recovery than PID. Although MPC requires slightly more computational time (0.83 s vs. 0.68 s), this trade-off is manageable in many control applications. Nonetheless, these findings are limited to simulation and do not yet account for real-time implementation challenges, such as tuning complexity, hardware limitations, or model uncertainty. Future work should address these practical concerns to assess MPC's viability for real-world DC motor systems.

1. Introduction

Electric motors play a critical role in modern industry, where they are used in more than 70% of equipment [1]. They are widely used in various applications, ranging from industrial automation, electric vehicles, robotics, home appliances, and precision manufacturing systems [2]-[4]. Based on the type of electrical current supplied, electric motors are generally categorized into two types: Direct Current (DC) motors and Alternating Current (AC) motors [5], [6]. Among these, DC motors, one of the earliest forms of electric motors, remain widely used due to their linear characteristics, simple modeling, and ease of speed and torque regulation [7]-[9]. Despite their control advantages, DC motors do have certain drawbacks, most notably in maintenance [10], [11]. The presence of brushes, which are prone to wear and tear, leads to higher maintenance costs compared to brushless AC motors. Nevertheless, DC motors remain popular in both research and industrial applications where precise control and fast response are essential, such as in robotic [12], [13], automotives [14]-[16], conveyor [17]-[19], and laboratory test setups [20]-[22].

In terms of control strategies, the Proportional–Integral–Derivative (PID) controller is still the most used method in industrial systems. According to [23], [24], approximately 90% of industrial control applications continue to rely on PID controllers due to their simplicity, ease of tuning, and broad applicability. However, PID controllers often struggle to handle constraints and model uncertainties, especially in systems with multivariable dynamics or significant time delays [25]. Moreover, the performance decreases when the system condition changes [26].

Model Predictive Control (MPC) has emerged as a powerful alternative to classical control techniques like PID, especially in applications where constraints on control inputs and system states must be explicitly addressed [27], [28]. It was first introduced in the 1970s and was adopted in many industries [29]. Originally developed for chemical process industries, MPC has since been successfully applied in robotics [30], [31], motor control [32]-[34], automotive systems [35]-[37], and other domains. MPC offers several benefits, such as its formulation in the time domain, inherent flexibility, capability to handle multiple objectives, suitability for multi-input multi-output (MIMO) systems, and straightforward integration of constraints and nonlinear dynamics [38]-[40]. Besides that, it also has weaknesses, such as high computational time due to online optimization [41], [42], and require accurate system modeling [43], [44].

MPC solves an optimization problem in real-time over a finite prediction horizon, using a system model to predict future behavior and determine optimal control actions. MPC falls under the broader category of optimal control and is closely related to Linear Quadratic Regulation (LQR). The key difference lies in their approach: LQR solves for a single optimal control law over the entire time horizon, while MPC solves a new optimization problem at each time step, adapting to real-time system conditions [45]. Moreover, MPC can accommodate constraints where the LQR cannot [44]. MPC is generally categorized into two types: linear and nonlinear [46]-[48]. Linear MPC uses a linear model, a quadratic cost function, and linear constraints, and is solved via quadratic programming. In contrast, nonlinear MPC incorporates nonlinear models, possibly non-quadratic cost functions, and nonlinear constraints, and requires nonlinear programming techniques.

The objective of this article is to provide a basic tutorial on Model Predictive Control using a DC motor speed control problem as a case study. Previous studies, such as in [49] give the MPC tutorial for complex systems. The authors in [50], give a tutorial on the MPC implementation related to the computer architecture with customization in the field programmable gate array (FPGA). While the application of MPC in the DC motor drive system has already been proven by [51]-[54].

The main contribution of this article lies in its educational and practical value, providing a step-by-step and intuitive tutorial on implementing MPC for DC motor speed control. By applying MPC to a simple and well-understood control task, this article aims to offer readers an accessible introduction to MPC principles, including system modeling, prediction horizon setup, constraint handling, and cost function formulation. The tutorial is intended for students, researchers, and engineers seeking a foundational understanding of MPC through a practical and intuitive example.

This paper is organized as follows. Section II presents the theoretical review of MPC. In section III, the design of MPC for speed control of the DC motor is presented. In section IV, the simulation results are discussed. Finally, the conclusion is in section V.

2. Model Predictive Control (MPC)

The basic structure of the MPC algorithm is depicted in Fig. 1. It shows that the MPC algorithm consists of a Model and an Optimizer. The model is used to get the predicted output. The difference between the reference signal and the predicted output is sent to the optimizer to get the future input.

The MPC works in the matrix domain. For more details, the prediction and optimization concept will be discussed in this section.

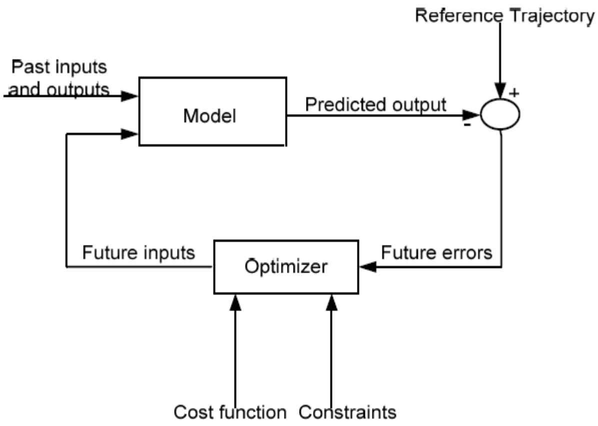

Fig. 1. Model predictive control (MPC) process [46]

2.1. Prediction Concept

Fig. 2 shows the prediction concept in MPC, specifically, how future process outputs are predicted, and control actions are calculated accordingly. Along the horizontal time axis, the diagram is divided into three regions: the past, the present (denoted by time step k), and the future (from $k + 1$ to $k + N_p$, where p is the prediction horizon). The green line represents the predicted output based on the system model and the current state information at time k. The dashed line represents the desired set-point or reference trajectory that the system aims to follow.

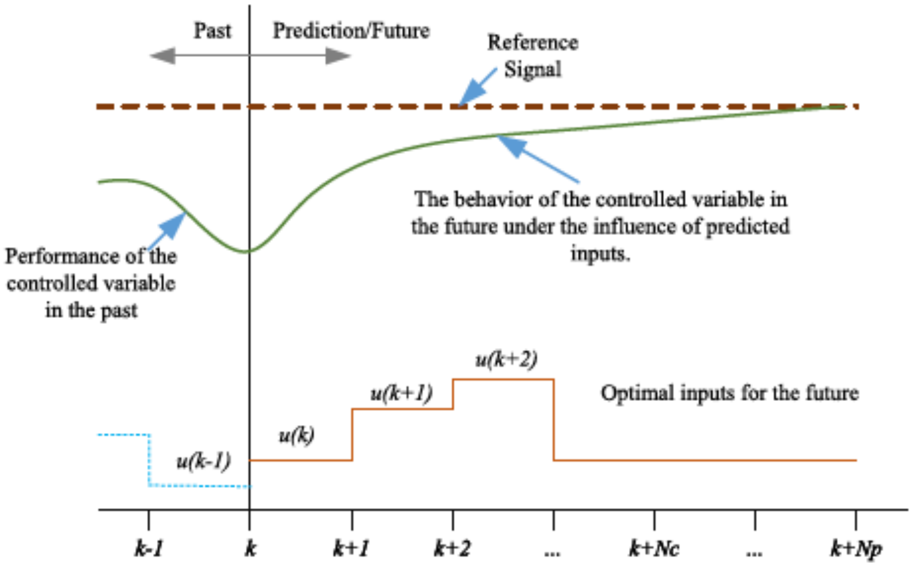

Fig. 2. Receding horizon concept in MPC [55]

Below the predicted outputs, the stepped line illustrates the control inputs or manipulated variables that the MPC algorithm optimizes. These inputs are calculated to minimize the deviation between the predicted outputs and the reference target, subject to system constraints. Although MPC computes a sequence of control actions for the entire prediction horizon, it only applies the first control input ($u(k)$) at the current time step. At the next step ($k + 1$), the process is repeated with updated system measurements, a concept known as the receding horizon strategy. This predictive and iterative nature allows MPC to anticipate future behavior, handle constraints, and optimize performance over time.

The next step state $(k+1)$ is calculated using (1), continuing for $x(k+2)$ state, etc. The same way is used to calculate the future output as in (3). For the N steps, the future output can be written as (4).

$$x(k+1) = Ax(k) + Bu(k) \tag{1}$$

$$\begin{aligned} x(k+2) &= Ax(k+1) + Bu(k+1) \\ &= A^2x(k) + ABu(k) + Bu(k+1) \end{aligned} \tag{2}$$
Etc.

$$\begin{aligned} y(k+1) &= Cx(k+1) \\ y(k+2) &= Cx(k+2) \end{aligned} \tag{3}$$
Etc.

$$\begin{aligned} Y &= \Phi \mathrm{x(k)} + \Gamma U \\ Y &= [y(k), y(k+1), \dots, y(k+N)]^T \\ U &= [u(k), u(k+1), \dots, u(k+N-1)]^T \end{aligned} \tag{4}$$

where Y, U, Φ, and Γ are predicted outputs for N steps, predicted input for N steps, matrix predicted state, and matrix predicted input, respectively.

2.2. Optimization Concept

After the prediction of future output is done, it is subtracted from the reference signal in the case of tracking control. The objective function of tracking control is in formula (5), the first term is related to tracking error, while the second term is related to the control effort required.

Since the MPC algorithm works in the matrix domain, the objective function in (5) is converted to the matrix domain as (6). Then, the weighting matrix Q for output error and R for input effort are added. Substitute (4) for (7) to get (8). Convert it into Quadratic Programming (QP) form to get (9) and rearrange to get (10). Lastly, the final objective function of MPC in the matrix form, which accommodates the prediction, is obtained as (11). H and f are quadratic cost matrices for input penalty and linear cost matrices for tracking error, respectively. With the objective function as in (11), the task of the optimization is to find the input (U) that can minimize the objective function. It can be written in the mathematical equation as in (12).

$$J(U) = \sum_{i=0}^{N-1} (y(k+i) - r(k+i))^2 + \sum_{i=0}^{N-1} u(k+i)^2 \tag{5}$$

$$J(U) = \| Y - R \|_Q^2 + \| U \|_R^2 \tag{6}$$

$$J(U) = (Y-R)^T Q (Y-R) + U^T R U \tag{7}$$

Substitute Y to get

$$J(U) = (\Phi x + \Gamma U - R)^T Q (\Phi x + \Gamma U - R) + U^T R U \tag{8}$$

Write in QP form

$$J(U) = U^T \Gamma^T Q \Gamma U + 2U^T \Gamma^T Q (\Phi x - R) + (\Phi x - R)^T Q (\Phi x - R) + U^T R U \tag{9}$$

$$J(U) = U^T (\Gamma^T Q \Gamma + R) U + 2U^T \Gamma^T Q (\Phi x - R) + (\Phi x - R)^T Q (\Phi x - R) \tag{10}$$

$$J(U) = U^T H U + 2f^T U + (\Phi x - R)^T Q (\Phi x - R)$$
$$H = \Gamma^T Q \Gamma + R \tag{11}$$
$$f = \Gamma^T Q (\Phi x - R)$$

$$U^* = \arg\min(U^T H U + 2f^T U + (\Phi x - R)^T Q (\Phi x - R)) \tag{12}$$

In the MPC optimization, constraints can be added. The constraints in the MPC are:

a) Input constraint (e.g., actuator saturation) : $u_{min} \leq u(k+i) \leq u_{max}$
b) Output constraint (e.g., system safety limit) : $y_{min} \leq y(k+i) \leq y_{max}$
c) States constraint (e.g., physical bounds) : $x_{min} \leq x(k+i) \leq x_{max}$

3. Design of MPC for Speed Control of DC Motor

This section discusses the step-by-step design of DC motor speed control using MPC, including the simulation setup.

3.1. Control Design

We follow a three-step approach to design the MPC for DC motor speed control, as illustrated in Fig. 3. The steps include modeling the DC motor, incorporating a state estimator, and solving the optimization problem inherent in the MPC formulation. Fig. 4 depicts the conceptual block diagram of the MPC-based speed control system with a Kalman Filter estimator. The MPC uses a prediction model and optimization solver to compute control inputs subject to constraints. The Kalman Filter estimates unmeasured states required for MPC operation based on output feedback.

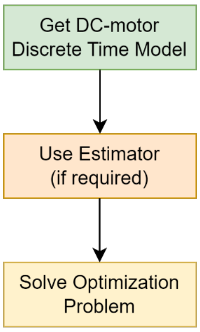

Fig. 3. Design flow of MPC for speed control of DC motor

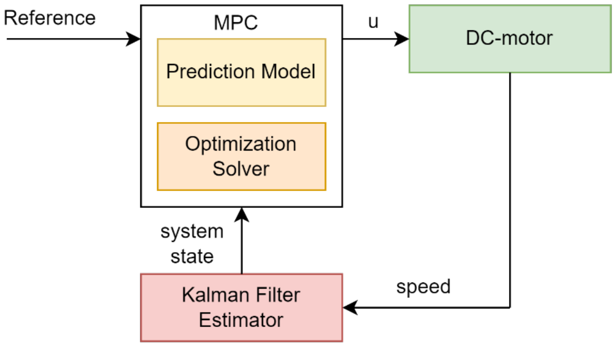

Fig. 4. Block diagram of the MPC-based speed control system with a Kalman Filter estimator

3.1.1. DC Motor Model

The DC motor is modeled using a discrete-time linear time-invariant (LTI) state-space representation adopted from [19], derived using the black-box system identification method. This modeling approach is commonly used in MPC applications due to its simplicity and compatibility with prediction-based control strategies [56]. The system matrices are defined in (13) as:

$$A = \begin{bmatrix} 0.91 & 0.04 \\ 1 & 0 \end{bmatrix}; \quad B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}; \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix}; \quad D = [0] \tag{13}$$

To ensure zero steady-state error for tracking control, an integrator is added to the system. The integrator in the discrete model is a pole at $z = 1$ for the transfer function model or an eigenvalue at $z = 1$ for the state space model. Based on (14), there is no eigenvalue of the DC motor equation in (13) with $z = 1$. Therefore, the system has no integrator, and hence, the integrator needs to be added. The integrator is appended via model augmentation to enhance closed-loop performance and guarantee asymptotic tracking. While this approach improves tracking, its robustness to modeling errors or disturbances will be further investigated in future work.

$$\begin{gathered} \det(zI - A) = 0 \\ \det\left(\begin{bmatrix} z - 0.91 & -0.04 \\ -1 & z \end{bmatrix}\right) = 0 \\ z(z - 0.91) - (-0.04)(-1) = z^2 - 0.91z - 0.04 \\ z = \frac{0.91 \pm \sqrt{0.91^2 + 4 \cdot 0.04}}{2} \\ z_1 = 0.952; \; z_2 = -0.042 \end{gathered} \tag{14}$$

Then we add an integrator as an augmented state (x_a) as

$$x_a = \begin{bmatrix} x \\ x_I \end{bmatrix} \tag{15}$$

$$A_a = \begin{bmatrix} A & 0 \\ -C & 1 \end{bmatrix}; \quad B_a = \begin{bmatrix} B \\ 0 \end{bmatrix}; \quad C_a = \begin{bmatrix} C & 0 \end{bmatrix} \tag{16}$$

$$A_a = \begin{bmatrix} 0.91 & 0.04 & 0 \\ 1 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}; \quad B_a = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}; \quad C_a = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \tag{17}$$

3.1.2. Estimator

MPC requires full state information for prediction. When not all states are measurable, an estimator is necessary. In this study, a Kalman Filter (KF) is used due to its widespread use in MPC and its optimality properties under Gaussian noise assumptions [57], [58]. This also refers to Linear Quadratic Gaussian (LQG), which combines LQR as state feedback control and KL as estimator. The KF will not be discussed in this study, because it is not the focus; instead, MATLAB is used to get the KF gain as show in Code 1.

Code 1. Kalman Filter gain calculation

```
[Kf, ~, ~] = dlqe(A, eye(2), C, Vd, Vn);
Kf = [0.1265; 0.0087]
```

In Code 1, $V_d = 0.01 * eye\,(2)$; and $V_n = 0.01$; are process noise covariance and measurement noise covariance, respectively.

3.1.3. Solve the Optimization Problem

The MPC cost function is defined as in (5). To get the minimum objective, optimal control input ($U *$) is required as in (12). To solve this optimization problem, Quadratic Programming (QP) is commonly used in MPC. For this basic tutorial, the analytical solution is used in which the optimal input is when the derivative is zero, as in (18). Therefore, the optimal control input is (19). Since the control input is in the matrix form, for the current input, only the first component is used (u_0) as in (20). Then, at the next step, repeat the process using the updated state $x(k+1)$.

$$\frac{dJ(U)}{dU} = 2HU + 2f = 0 \tag{18}$$

$$U^* = -H^{-1}f \tag{19}$$

$$u(k) = u_0 \tag{20}$$

In this tutorial implementation, input constraints of ±12 VDC (based on motor specs) are applied using a saturation function:

$$u_{min} \leq u \leq u_{max} \tag{21}$$

Although this “clipping” approach does not handle constraints during optimization, it simplifies implementation. For a more rigorous treatment, constrained QP solvers like quadprog should be used to enforce constraints within the optimization loop.

3.2. Simulation Design

The proposed control system is evaluated through simulation and compared to a standard PID controller. Fig. 5 shows the Simulink block diagram, with the DC motor modeled using the state-space block. Two input scenarios are considered: a step input and a reference-tracking signal generated via Signal Builder. The use of Simulink provides a practical environment that closely resembles real-time hardware systems. The MPC code is encapsulated in the MPC block and written as shown in Code 2.

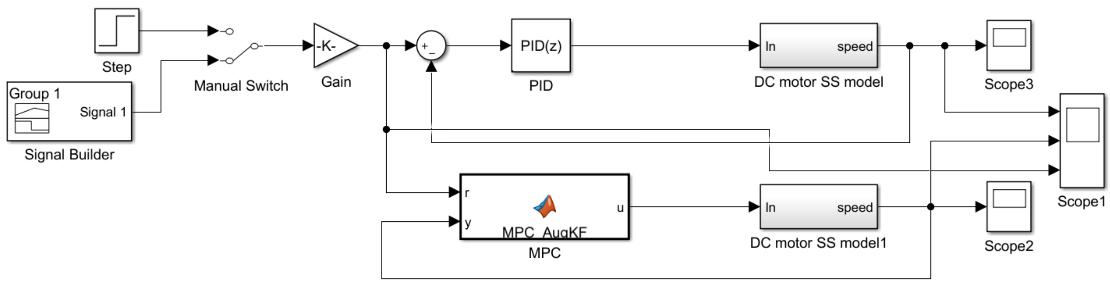

Fig. 5. Simulink model of control system

The MPC code in Code 2 starts from the constants and system matrices definition. Next, the augmented model is developed to add an integrator in the system model. The estimator uses KF with precomputed gain, and then the augmented state is calculated. Next, a prediction model is developed with prediction matrices. Finally, the QP problem is solved, and the KF is updated.

3.3. Parameter Tuning and Performance Metric

The performance of MPC in the speed control of the DC motor will be compared to that of the PID. In the MPC method, the tuned parameters are the prediction horizon (N), cost matrices (Q and R), and Kalman Filter gain. All parameters are tuned manually. More details about the Kalman Filter gain are already explained in Section 3.1.2. All the final parameters value of MPC is listed in Code 2. On the other hand, the PID gain parameters (Kp, Ki, and Kd) are derived from MATLAB auto-tuning. The final PID gain used is $Kp = 0.08$, $Ki = 0.25$, and $Kd = 0.16$.

The integral absolute error (IAE), which is the accumulated absolute error at the end of simulation time, will be used as the performance metric. The IAE formula is in (22) [59]. The control tolerance used is ± 1%. The other parameters, such as settling time, overshoot, undershoot, and computational time, are also used in the analysis.

$$IAE = \int_0^t |e(t)| dt \quad (22)$$

Code 2. MPC control method

```
function u = MPC_AugKF(y, r)
% === Constants ===
nx = 2;           % number of states
ni = 1;           % integrator
nxa = nx + ni;
nu = 1;
ny = 1;
N = 10;          % prediction horizon
% === System matrices ===
A = [0.91 0.04; 1.00 0];
B = [1;0];
C = [1 0];
% === Augmented model ===
Aa = [A, zeros(nx, ni); -C, 1];
Ba = [B; 0];
Ca = [C, 0];
 % === Cost weights ===
Q = eye(N * ny) * 20;
R = eye(N * nu) * 5;
% === Persistent variables ===
persistent x_hat x_i
if isempty(x_hat)
    x_hat = zeros(nx,1);
    x_i = 0;
end
% === Precomputed Kalman Gain ===
Kf = [0.1265; 0.0087];
% === Update integrator ===
x_i = x_i + (r - y);
% === Augmented state ===
x_a = [x_hat; x_i];
% === Input constraint ===
umax = 12;  umin=-12;
% === Prediction model ===
F = zeros(N * ny, nxa); %F for phi
G = zeros(N * ny, N * nu); %G for
gamma
ref = repmat(r, N * ny, 1);
Ai = eye(nxa);
for i = 1:N
    Ai = Ai * Aa;
    F(i,:) = Ca * Ai;
    for j = 1:i
        Aij = eye(nxa);
        for k = 1:(i-j)
            Aij = Aij * Aa;
        end
        G(i,j) = Ca * Aij * Ba;
    end
end
% === QP problem ===
H = G' * Q * G + R;
f = G' * Q * (F * x_a - ref);
u_opt = -H \ f;
u = u_opt(1);  % apply only first
input
uk  =  max(min(u,  umax),  umin);
%constraint
u = uk;
% === Kalman Filter update ===
x_hat = A * x_hat + B * u + Kf *
(y - C * x_hat);
end
```

4. Results and Discussion

4.1. Unit Step Response

The unit step response is the basic test to determine the performance of the controller [60]. Fig. 6 shows the step response of the system, which is PID and MPC control. It shows that MPC is better than PID in terms of settling time and overshoot. Both controller responses have zero steady-state errors. Detailed controller performance parameters can be seen in Table 1.

Table 1 presents a comparative analysis of unit step response performance between PID and MPC controllers. The results demonstrate a clear advantage of MPC over PID across all evaluated metrics. MPC significantly reduces the settling time from 24.5 seconds (PID) to 6.1 seconds, representing a 75% improvement in response speed. Additionally, MPC achieves a much lower overshoot of 1.5% compared to 6.4% for PID, indicating better transient response and improved system stability. The IAE, which measures overall tracking accuracy, is also considerably lower with MPC (191.0) than with PID (374.2), suggesting more accurate setpoint following. Although MPC requires slightly more computational time (0.83 s) compared to PID (0.68 s), the marginal increase is justifiable considering the substantial performance gains in speed, stability, and accuracy. These findings highlight MPC's superior control characteristics and reinforce its suitability for applications requiring precise and responsive behavior.

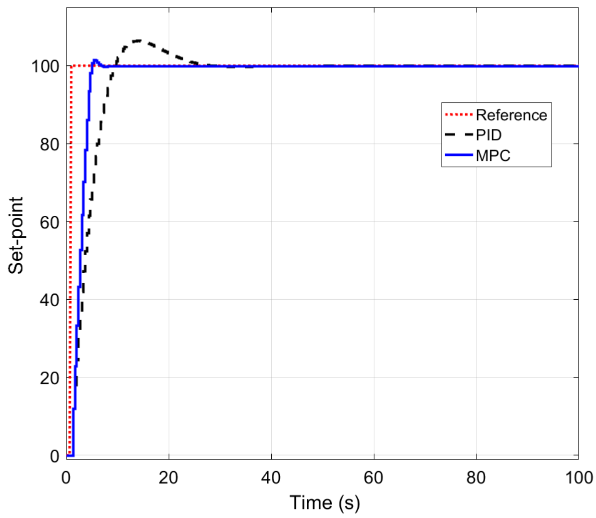

Fig. 6. Performance of unit step response

Table 1. Unit Step Performance

Control method	Settling time (s)	Overshoot (%)	IAE	Computational Time
PID	24.5	6.4	374.2	0.68
MPC	6.1	1.5	191.0	0.83

4.2. Disturbance Response

To evaluate the controller's response to system disturbances, an input disturbance is applied. The disturbance is added to the control signal before entering the DC motor, with an amplitude of 5 and a duration of 1 second. The resulting speed response is depicted in Fig. 7. It shows that the speed of both controllers increases during the disturbance, but once the disturbance ends, the system controlled by MPC returns to the reference speed more quickly and with minimal undershoot. In contrast, the system using the PID controller shows a slower recovery and a larger undershoot. The quantitative results are summarized in Table 2.

Table 2 presents a comparative analysis of the disturbance response performance between PID and MPC controllers. The results highlight the superior capability of MPC in handling disturbances. While the overshoot values for both controllers are comparable (12.45% for PID and 12.39% for

MPC), the MPC controller demonstrates a significantly lower undershoot of 1.10%, compared to 2.85% for the PID controller. This indicates that MPC is more effective in minimizing deviation below the setpoint after a disturbance occurs. Additionally, the IAE, which quantifies the total deviation from the setpoint, is much lower for MPC (205.70) than for PID (428.70). This substantial reduction in IAE further confirms that MPC provides more accurate and stable performance under disturbance conditions.

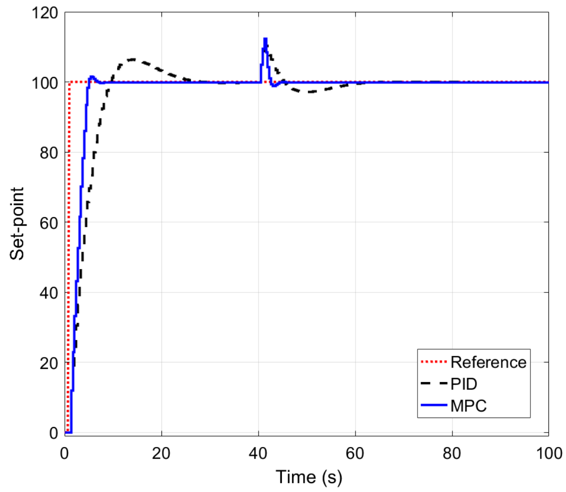

Fig. 7. Performance of disturbance response

Table 2. Disturbance response performance

Control method	Overshoot (%)	Undershoot (%)	IAE
PID	12.45	2.85	428.70
MPC	12.39	1.10	205.70

4.3. Signal Tracking

The next test is signal tracking. In this test, setpoint changes are given to the system. Fig. 8 and Table 3 collectively illustrate the signal tracking performance of PID and MPC. The figure shows the response of each controller to a varying reference signal. Visually, the MPC (blue line) closely follows the reference trajectory (red dotted line) with minimal delay and overshoot, whereas the PID controller (black dashed line) exhibits noticeable lag and overshoot, especially at points where the set-point changes abruptly.

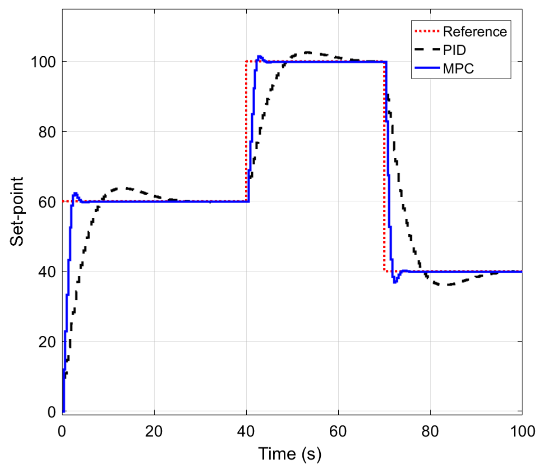

Fig. 8. Performance of signal tracking

This visual interpretation is quantitatively supported by Table 3, which presents the IAE for both methods. MPC achieves a significantly lower IAE value of 166.8 compared to PID's 596.6. The lower

IAE confirms that MPC offers better tracking accuracy, responding more effectively to reference changes while minimizing deviation over time. Overall, both the figure and table highlight MPC's superior performance in terms of precision, responsiveness, and stability during dynamic signal tracking.

Table 3. Signal tracking performance

Control method	IAE
PID	596.6
MPC	166.8

5. Conclusion

This tutorial paper introduced the basic implementation of MPC for DC motor speed control, with PID used as a benchmark to highlight MPC's advantages. Through three simulation-based tests, unit step response, input disturbance rejection, and signal tracking, the effectiveness of MPC was demonstrated in comparison to the widely used PID controller. In the step response test, MPC achieved a 75% reduction in settling time, a 76.6% reduction in overshoot, and a 49% improvement in tracking accuracy (IAE), demonstrating superior transient performance. Although MPC requires slightly more computational time (0.83 s) compared to PID (0.68 s). In the disturbance response test, MPC yielded lower undershoot and a 52% improvement in IAE, confirming its better recovery and disturbance attenuation. Finally, in the signal tracking scenario, MPC closely followed dynamic reference changes with a 72% lower IAE than PID, indicating greater adaptability and precision. Despite these promising results, the study is limited to simulation and does not address practical implementation aspects such as real-time performance, hardware constraints, or robustness under modeling uncertainties. These limitations suggest that further work is necessary before MPC can be widely adopted in real-world DC motor systems.

Author Contribution: All authors contributed equally to the main contributor to this paper. All authors read and approved the final paper.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

[1] B. K. Bose, "Global energy scenario and impact of power electronics in 21st century," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 7, pp. 2651–2663, 2012, https://doi.org/10.1109/TIE.2012.2203771.

[2] D. Kumar and S. J. Mija, "Design and performance evaluation of robust SMC schemes for speed control of DC motor," in *Proceedings of the 2014 IEEE International Conference on Advanced Communication Control and Computing Technologies (ICACCCT)*, 2014, pp. 88–92, https://doi.org/10.1109/ICACCCT.2014.7019235.

[3] H. Maghfiroh, O. Wahyunggoro, and A. I. Cahyadi, "Optimal sizing and improved low-pass filter energy management for hybrid energy storage electric vehicles," *IEEE Access*, vol. 12, pp. 180926–180938, 2024, https://doi.org/10.1109/ACCESS.2024.3509481.

[4] M. Tran, M. Trinh, N. Do, T. Phan, T. C. Ngo, and V. Nguyen, "Experimental swing-up control of advanced sliding and energy-based modes for Pendubot," *Journal of Fuzzy Systems and Control*, vol. 3, no. 1, pp. 51–56, 2025, https://doi.org/10.59247/jfsc.v3i1.277.

[5] H. Maghfiroh, I. Iftadi, and A. Sujono, "Speed control of induction motor using LQG," *Journal of Robotics and Control*, vol. 2, no. 6, pp. 565–570, 2021, https://doi.org/10.18196/jrc.26138.

[6] P. N. Bhatt, H. Mehar, and M. Sahajwani, “Electrical motors for electric vehicle – a comparative study,” in *Proceedings of the Recent Advances in Interdisciplinary Trends in Engineering & Applications (RAITEA)*, 2019, https://doi.org/10.2139/ssrn.3364887.

[7] H. Maghfiroh, M. Nizam, and S. Praptodiyono, “PID optimal control to reduce energy consumption in DC-drive system,” *International Journal of Power Electronics and Drive Systems*, vol. 11, no. 4, pp. 2164–2172, 2020, https://doi.org/10.11591/ijpeds.v11.i4.pp2164-2172.

[8] Q. Ariyansyah and Alfian Ma’arif, “DC motor speed control with proportional integral derivative (PID) control on the prototype of a robot,” *Journal of Fuzzy Systems and Control*, vol. 1, no. 1, pp. 18–24, 2023, https://doi.org/10.59247/jfsc.v1i1.26.

[9] M. Alfiyan, R. D. Puriyanto, U. A. Dahlan, and C. Author, “Mecanum 4 omni wheel directional robot design system using PID method,” *Journal of Fuzzy Systems and Control*, vol. 1, no. 1, pp. 6–13, 2023, https://doi.org/10.59247/jfsc.v1i1.27.

[10] R. Rikwan and A. Ma’arif, “DC motor rotary speed control with Arduino UNO based PID control,” *Control Systems Optimization Letters*, vol. 1, no. 1, pp. 17–31, 2023, https://doi.org/10.59247/csol.v1i1.6.

[11] H. Maghfiroh, M. R. Subeno, M. R. Darmawan, and R. Prihananto, “A survey on traction motor and its prototyping method for electric vehicle application,” *Journal of Electrical Electronic Information and Communication Technology*, vol. 5, no. 1, p. 21, May 2023, https://doi.org/10.20961/jeeict.5.1.71317.

[12] N. X. Chiem and N. C. B. Nguyen, “Design of embedded control system with fuzzy controller and nonlinear controller for the line follower robot,” *Journal of Fuzzy Systems and Control*, vol. 3, no. 2, pp. 122–127, 2025, https://doi.org/10.59247/jfsc.v3i2.303.

[13] M. Z. Hussin, J. Jalani, M. H. Powdzi, S. M. Rejab, and M. K. Ishak, “Smart robot cleaner using Internet of Things,” *Journal of Advanced Research in Applied Sciences and Engineering Technology*, vol. 46, no. 1, pp. 175–186, 2025, https://doi.org/10.37934/araset.46.1.175186.

[14] A. Chatterjee, “ANFIS-based fault detection in brushed and brushless DC motors: A hybrid intelligence approach,” *Journal of Fuzzy Systems and Control*, vol. 3, no. 2, pp. 149–154, 2025, https://doi.org/10.59247/jfsc.v3i2.312.

[15] A. H. Hameed and S. A. Al-Samarraie, “Ultimate bounded observer-based control of electrical vehicle driven by DC motor system with unmatched load torque,” *Advances in Mechanical Engineering*, vol. 16, no. 10, pp. 1–14, 2024, https://doi.org/10.1177/16878132241273549.

[16] B. Satria, H. Alam, D. Erpandi, and A. D. Trg, “A DC motor speed control for two wheeled vehicles with PWM,” *Jurnal Scientia*, vol. 13, no. 4, pp. 1469–1476, 2024, https://infor.seaninstitute.org/index.php/pendidikan/article/view/2631.

[17] D. Sonny and R. D. Puriyanto, “Implementation of DC motor PID control on conveyor for separating potato seeds by weight,” *International Journal of Robotics and Control Systems*, vol. 1, no. 1, pp. 15–26, 2021, https://doi.org/10.31763/ijrcs.v1i1.221.

[18] H. Maghfiroh, R. Martfuadi, and M. H. Ibrahim, “Optimal energy control of DC motor using fuzzy logic controller to supervise PID control,” *AIP Conference Proceedings*, vol. 2217, no. 1, p. 030060, Apr. 2020, https://doi.org/10.1063/5.0000617.

[19] H. Maghfiroh, M. Gunawan, and M. Anwar, “Optimal energy control of DC-drive conveyor using LQR method,” *AIP Conference Proceedings*, vol. 2217, no. 1, p. 030145, Apr. 2020, https://doi.org/10.1063/5.0000619.

[20] T. Le, N. Nguyen, P. Le, M. Bui, T. Nguyen, and C. Tran, “LQR controller based on BAT algorithm for rotary double parallel inverted pendulum,” *Journal of Fuzzy Systems and Control*, vol. 3, no. 2, pp. 109–116, 2025, https://doi.org/10.59247/jfsc.v3i2.304.

[21] V. Tran, T. Le, D. Hoang, V. Nguyen, N. Ho, and T. Do, “Backstepping control for ball and beam: Simulation and experiment,” *Journal of Fuzzy Systems and Control*, vol. 3, no. 1, pp. 30–38, 2025, https://doi.org/10.59247/jfsc.v3i1.275.

[22] N. R. Setiawan, A. Ma, and N. S. Widodo, “DC motor controller using full state feedback,” *Control Systems Optimization Letters*, vol. 1, no. 1, pp. 7–11, 2023, https://doi.org/10.59247/csol.v1i1.3.

[23] S. M. Rakhtala and E. S. Roudbari, “Application of PEM fuel cell for stand-alone based on a fuzzy PID control,” *Bulletin of Electrical Engineering and Informatics*, vol. 5, no. 1, pp. 45–61, 2016, https://doi.org/10.11591/eei.v5i1.521.

[24] 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, 2005, https://doi.org/10.1109/TCST.2005.847331.

[25] K. J. Åström and T. Hägglund, “The future of PID control,” *Control Engineering Practice*, vol. 9, pp. 1163–1175, 2001, https://doi.org/10.1016/S0967-0661(01)00062-4.

[26] H. Maghfiroh, O. Wahyunggoro, A. I. Cahyadi, and S. Praptodiyono, “PID-hybrid tuning to improve control performance in speed control of DC motor based on PLC,” in *Proc. 3rd Int. Conf. Instrumentation, Control and Automation (ICA)*, 2013, pp. 233–238, https://doi.org/10.1109/ICA.2013.6734078.

[27] T. Englert and K. Graichen, “Nonlinear model predictive torque control and setpoint computation of induction machines for high performance applications,” *Control Engineering Practice*, vol. 99, p. 104415, 2020, https://doi.org/10.1016/j.conengprac.2020.104415.

[28] F. Wang, X. Mei, J. Rodriguez, and R. Kennel, “Model predictive control for electrical drive systems—an overview,” *CES Transactions on Electrical Machines and Systems*, vol. 1, no. 3, pp. 219–230, 2017, https://doi.org/10.23919/TEMS.2017.8086100.

[29] J. B. Rawlings, D. Q. Mayne, and M. M. Diehl, *Model Predictive Control: Theory, Computation, and Design*, 2nd ed. Madison, WI, USA: Nob Hill Publishing, 2017, https://books.google.co.id/books?id=MrJctAEACAAJ.

[30] J. S. Saputro, P. H. Rusmin, and A. S. Rochman, “Design and implementation of trajectory tracking motion in mobile robot skid steering using model predictive control,” in *Proc. IEEE 8th Int. Conf. System Engineering and Technology*, 2018, pp. 73–78, https://doi.org/10.1109/ICSEngT.2018.8606361.

[31] H. Nguyen, M. Kamel, K. Alexis, and R. Siegwart, “Model predictive control for micro aerial vehicles: a survey,” in *Proc. European Control Conference*, 2021, pp. 1556–1563, https://doi.org/10.23919/ECC54610.2021.9654841.

[32] A. Ramelan, A. S. Rohman, and A. Kelana, “Embedded position control of permanent magnet synchronous motor using model predictive control,” in *Asia Conference on Mechanical Engineering and Aerospace Engineering*, 2018, vol. 04007, pp. 2–6, https://doi.org/10.1051/matecconf/201819804007.

[33] S. Bolognani and S. Bolognani, “Design and implementation of model predictive control for electrical motor drives,” *IEEE Transactions on Industrial Electronics*, vol. 56, no. 6, pp. 1925–1936, 2009, https://doi.org/10.1109/TIE.2008.2007547.

[34] Y. Zhang, B. Xia, H. Yang, and J. Rodriguez, "Overview of model predictive control for induction motor drives," *Chinese Journal of Electrical Engineering*, vol. 2, no. 1, pp. 62–76, 2016, https://doi.org/10.23919/CJEE.2016.7933116.

[35] C. Jia, J. Cui, W. Qiao, and L. Qu, "Real-time model predictive control for battery-supercapacitor hybrid energy storage systems using linear parameter-varying models," *IEEE Journal of Emerging and Selected Topics in Power Electronics*, vol. 11, no. 1, pp. 251–263, 2023, https://doi.org/10.1109/JESTPE.2021.3130795.

[36] H. Chen, R. Xiong, C. Lin, and W. Shen, “Model predictive control based real-time energy management for hybrid energy storage system,” *CSEE Journal of Power and Energy Systems*, vol. 7, no. 4, pp. 862–874, 2021, https://doi.org/10.17775/CSEEJPES.2020.02180.

[37] Y. Huang, H. Wang, A. Khajepour, H. He, and J. Ji, “Model predictive control power management strategies for HEVs: a review,” *Journal of Power Sources*, vol. 341, pp. 91–106, 2017, https://doi.org/10.1016/j.jpowsour.2016.11.106.

[38] G. Mirzaeva and Y. Mo, “Model predictive control for industrial drive applications,” *IEEE Transactions on Industry Applications*, vol. 59, no. 2, pp. 7897–7907, 2023, https://doi.org/10.1109/TIA.2023.3299887.

[39] P. Huynh, K. Le, T. Nguyen, H. Tran, and S. Dang, "A study of adaptive model predictive control for rotary inverted pendulum," *Journal of Fuzzy Systems and Control*, vol. 3, no. 2, pp. 98–103, 2025, https://doi.org/10.59247/jfsc.v3i2.302.

[40] C. Li, Q. Meng, and T. Shi, "A review on model predictive control strategies for AC motor drives," *IET Electric Power Applications*, vol. 18, no. 11, pp. 1584–1604, 2024, https://doi.org/10.1049/elp2.12510.

[41] S. Kouro, M. A. Perez, J. Rodriguez, A. M. Llor, and H. A. Young, "Model predictive control: MPC's role in the evolution of power electronics," *IEEE Industrial Electronics Magazine*, vol. 9, no. 4, pp. 8–21, 2015, https://doi.org/10.1109/MIE.2015.2478920.

[42] G. A. Papafotiou, G. D. Demetriades, and V. G. Agelidis, "Technology readiness assessment of model predictive control in medium- and high-power applications," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 9, pp. 5807–5815, 2016, https://doi.org/10.1109/TIE.2016.2521350.

[43] M. G. Forbes, R. S. Patwardhan, H. Hamadah, and R. B. Gopaluni, "Model predictive control in industry: Challenges and opportunity," *IFAC-PapersOnLine*, vol. 48, no. 8, pp. 531–538, 2015, https://doi.org/10.1016/j.ifacol.2015.09.022.

[44] J. L. Garriga and M. Soroush, "Model predictive control tuning methods: A review," *Industrial & Engineering Chemistry Research*, vol. 49, pp. 3505–3515, 2010, https://doi.org/10.1021/ie900323c.

[45] P. Huynh, M. Nguyen, N. Pham, H. Duong, and H. Nguyen, "Model predictive control for rotary inverted pendulum: Simulation and experiment," *Journal of Fuzzy Systems and Control*, vol. 2, no. 3, pp. 215–222, 2024, https://doi.org/10.59247/jfsc.v2i3.263.

[46] P. E. Orukpe, "Model Predictive Control Fundamentals," *Niger. J. Technol.*, vol. 31, no. 2, pp. 139-148, 2012, https://www.ajol.info/index.php/njt/article/view/123569.

[47] W. Li, X. Zhang, Y. Wang, and S. Xie, "Comparison of linear and nonlinear model predictive control in path following of underactuated unmanned surface vehicles," *Journal of Marine Science and Engineering*, vol. 12, no. 4, p. 575, 2024, https://doi.org/10.3390/jmse12040575.

[48] T. Ghandriz, B. Jacobson, P. Nilsson, and L. Laine, "Trajectory-following and off-tracking minimisation of long combination vehicles: A comparison between nonlinear and linear model predictive control," *International Journal of Vehicle Mechanics and Mobility*, vol. 62, no. 2, pp. 1–27, 2024, https://doi.org/10.1080/00423114.2022.2164513.

[49] E. N. Hartley, "A tutorial on model predictive control for spacecraft rendezvous," in *Proc. 2015 European Control Conference (ECC)*, 2015, pp. 1355–1361, https://doi.org/10.1109/ECC.2015.7330727.

[50] G. A. Constantinides, "Tutorial paper: Parallel architectures for model predictive control," in *Proc. 2009 European Control Conference (ECC)*, 2009, pp. 138–143, https://doi.org/10.23919/ECC.2009.7074393.

[51] L. E. J. Alkurawy and N. Khamas, "Model predictive control for DC motors," in *Proc. 2018 1st International Scientific Conference of Engineering Sciences – 3rd Scientific Conference of Engineering Science (ISCES)*, 2018, pp. 56–61, https://doi.org/10.1109/ISCES.2018.8340528.

[52] S. Sahoo, B. Subudhi, and G. Panda, "Optimal speed control of DC motor using linear quadratic regulator and model predictive control," in *Proc. 2015 Int. Conf. Energy, Power and Environment: Towards Sustainable Growth (ICEPE)*, 2015, pp. 1–5, https://doi.org/10.1109/EPETSG.2015.7510130.

[53] H. Pandya, D. R. Vyas, P. S. Thakar, A. Markana, and S. Prajapati, "Improved robust and optimal performance of DC servo motor using model predictive control with implementation," *Advanced Control for Applications: Engineering and Industrial Systems*, vol. 7, no. 3, pp. 1-12, 2025, https://doi.org/10.1002/adc2.70024.

[54] A. H. Malik, F. Arshad, and A. A. Memon, "Design of advanced controllers for speed control of DC motor," *Proceedings of the Pakistan Academy of Sciences: A. Physical and Computational Sciences*, vol. 61, no. 1, pp. 89–97, 2024, https://doi.org/10.53560/PPASA(61-1)828.

[55]] M. B. Shahid, W. Jin, M. A. Abbasi, A. R. B. Husain, H. M. Munir, M. Hassan, A. Flah, A. S. E. Souissi, and T. A. H. Alghamdi, "Model predictive control for energy efficient AC motor drives: An overview," *IET Electric Power Applications*, vol. 18, no. 12, pp. 1894–1920, 2024, https://doi.org/10.1049/elp2.12517.

[56] A. A. Bature, S. Buyamin, M. N. Ahmad, M. Muhammad, and A. M. Abdullahi, "Sensorless position control of DC motor using model predictive controller," *Jurnal Teknologi (Sciences & Engineering)*, vol. 77, no. 12, pp. 17-21, Nov. 2015, https://doi.org/10.11113/jt.v77.6303.

[57] M. Liu, W. Chen, J. Huang, and Y. Ning, "Kalman filter-based model predictive control for an adaptive cruise control system considering measurement noise," *SAE International Journal of Connected and Automated Vehicles*, vol. 3, no. 1, pp. 1–15, 2020, https://doi.org/10.4271/12-03-01-0005.

[58] S. A. Davari, S. Azadi, F. Flores-Bahamonde, F. Wang, P. Wheeler, and J. Rodriguez, "Compensating the measurement error in model-free predictive control of induction motor via Kalman filter-based ultra-local model," *IEEE Transactions on Power Electronics*, vol. 39, no. 12, pp. 15811–15821, 2024, https://doi.org/10.1109/TPEL.2024.3443134.

[59] A. Rafi, A. Tahtawi, S. Yahya, P. Elbizzar, and S. M. Ilman, "Speed control of three-phase 1.5 kW induction motor using VSD LS SV015IG5A-2 with proportional integral anti-windup method," *Journal of Fuzzy Systems and Control*, vol. 2, no. 3, pp. 140–146, 2024, https://doi.org/10.59247/jfsc.v2i3.242.

[60] H. Maghfiroh, A. Sujono, M. Ahmad, and C. H. B. Apribowo, "Basic tutorial on sliding mode control in speed control of DC-motor," *Journal of Electrical, Electronic Information, and Communication Technology*, vol. 2, no. 1, pp. 1–4, 2020, https://doi.org/10.20961/jeeict.2.1.41354.