International Journal of Advances in Engineering and Management (IJAEM)
Volume 7, Issue 06 June 2025, pp: 360-368 www.ijaem.net ISSN: 2395-5252

Application of Long Short-Term Memory (LSTM) Networks for Adaptive PID Parameter Prediction in Dynamic DC Motor Speed Control

1st Hatim O. Elhassan, 2nd M. I. Shukri
Alsalama College of Science & Technology Khartoum, Khartoum North, Sudan
Alsalama College of Science & Technology Khartoum, Khartoum North, Sudan

Date of Submission: 10-06-2025 Date of Acceptance: 20-06-2025

ABSTRACT: This paper investigates the application of Long Short-Term Memory (LSTM) networks for adaptive PID parameter prediction in dynamic DC motor speed control. The study aims to enhance precision, adaptability, and robustness beyond conventional methods by leveraging LSTM's capability to model time-dependent behaviours. Methodology involves designing and simulating LSTM-based controllers that capture temporal dependencies for rapid response under varying conditions. A synthesized dataset of optimal PID parameters, optimized for diverse operating conditions, trains the LSTM model using error, integral, and derivative inputs. Key findings show LSTM consistently achieves the fastest rise and settling times across scenarios, demonstrating superior dynamic response and rapid stabilization. While computationally intensive, LSTM offers strong generalization capabilities. This research significantly advances intelligent control systems, proving LSTM's potential for high-speed, dynamic applications in areas like autonomous vehicles and industrial automation.

KEYWORDS: DC Motor Control, Artificial Neural Network (ANN), Long Short-Term Memory (LSTM), PID Control, Parameter Prediction, Speed Regulation.

I. INTRODUCTION

Direct current (DC) motors are fundamental components of electromechanical systems, widely used in robotics, automotive systems, aerospace, and industrial automation due to their ability to convert electrical energy into precise mechanical motion [1]. Their reliability and versatility have sustained their relevance despite the rise of alternating current (AC) motors [2, 3]. Effective control strategies are essential for optimizing DC motor performance, particularly under varying loads and nonlinear dynamics. Traditional Proportional-Integral-Derivative (PID) controllers, while commonly used for regulating speed and torque, often fail to adapt to real-world complexities such as load variations, friction, and thermal effects due to their reliance on fixed parameters and assumptions of linearity [4]. Recent advancements in adaptive control techniques, including those leveraging Long Short-Term Memory (LSTM) networks, offer promising solutions to these limitations. LSTM networks excel in modelling time-dependent behaviours and capturing temporal dependencies, making them ideal for predictive control in dynamic environments [5]. Unlike conventional methods, LSTM-based controllers can continuously learn from system feedback and adjust control parameters in real time, thereby enhancing precision, adaptability, and robustness [6]. Despite these advantages, there remains a significant research gap in the application of LSTM networks specifically for adaptive PID control in DC motor speed regulation. While some studies have explored general artificial neural network (ANN) applications, systematic investigations into LSTM's capabilities in handling nonlinearities and disturbances under real-time conditions remain limited [7]. This raises a critical research question: How to make an artificial neural network model implementation of adaptive PID control of DC motor speed that is reliable and robust to maintain precise control under various disturbances?

International Journal of Advances in Engineering and Management (IJAEM)
Volume 7, Issue 06 June 2025, pp: 360-368 www.ijaem.net ISSN: 2395-5252

This paper addresses this challenge by designing and simulating an LSTM-based controller using Python's machine learning toolbox. The study evaluates its performance against key metrics—rise time, settling time, and overshoot—to demonstrate improvements over traditional PID approaches. The contribution lies in advancing DC motor control through LSTM strategies that offer superior temporal modelling for dynamic speed regulation. Limitations include reliance on simulation-based analysis, use of an idealized motor model, and testing under controlled scenarios, which may not fully capture real-world complexities [6].

II. DC MOTOR

A Direct Current (DC) motor is an electromechanical device that converts electrical energy into mechanical motion [8]. Its construction consists of two primary components: the stator and the rotor. The stator, the stationary part, contains poles excited by a direct current to produce a magnetic field. The rotor, the rotating component, features a laminated iron core with slots housing coils. These coils are connected in series and interact with the magnetic field to generate motion [9].

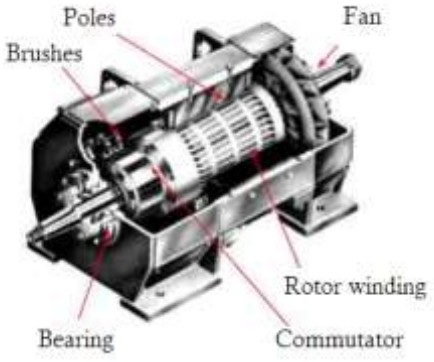

Fig 1. Dc Motor Roter Construction

The principle of operation of a DC motor is rooted in electromagnetic induction and torque generation. Electromagnetic induction occurs when the rotor rotates within the stator's magnetic field, causing the armature coils to cut through magnetic flux lines. This induces an electromotive force (EMF), known as back EMF:

$$E_b = K_e \omega \#(1)$$

Where E_b is the back EMF, K_e is the motor's back EMF constant, and ω is the angular velocity of the rotor.

Torque generation arises from the interaction between the magnetic field and current flowing through the armature windings. The resulting torque drives the rotor and is expressed as:

$$T = K_t I_a \#(2)$$

where T is the torque, K_t is the torque constant, and I_a is the armature current.

The voltage equation for the armature circuit is given by:

$$V_a = E_b + I_a R_a + L_a \frac{dI_a}{dt} \#(3)$$

In steady-state conditions ($\frac{dI_a}{dt} = 0$), this simplifies to:

$$V_a = K_e \omega + I_a R_a \#(4)$$

These equations collectively describe how applied voltage and current influence motor speed and torque, underpinning its operational dynamics [10].

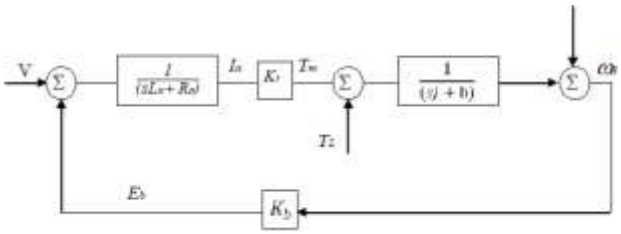

Fig 2. DC Motor Block Diagram

III. PID CONTROLLER

The Proportional-Integral-Derivative (PID) controller is an important of control engineering, widely used for its simplicity and effectiveness in regulating dynamic systems such as DC motors.A PID controller minimizes the error between the desired setpoint (ω_{set})and the actual output $(\omega(t))$ by applying a weighted sum of three terms: proportional, integral, and derivative. The control signal $(V(t))$, which corresponds to the applied voltage in the case of a DC motor [11], is computed as:

$$V(t) = K_p \cdot e(t) + K_i \cdot \int e(t)\, dt + K_d \cdot \frac{de(t)}{dt} \#(5)$$

Where:

- $e(t) = \omega_{set} - \omega(t)$: Error between the set speed and the current speed.
- K_p: Proportional gain, which directly reduces the error.
- K_i: Integral gain, which eliminates steady-state error by accumulating past errors over time.

International Journal of Advances in Engineering and Management (IJAEM)
Volume 7, Issue 06 June 2025, pp: 360-368 www.ijaem.net ISSN: 2395-5252

K_d: Derivative gain, which minimizes overshoot and oscillations by predicting future trends based on the rate of change of the error [11].

IV. LONG SHORT-TERM MEMORY (LSTM)

Long Short-Term Memory (LSTM)Long Short-Term Memory (LSTM) networks are a specialized class of recurrent neural networks (RNNs) designed to overcome the vanishing gradient problem inherent in traditional RNNs, enabling them to learn long-term dependencies in sequential data [12]. Their architecture is structured around memory blocks called cells, which regulate information flow through three key components: input, output, and forget gates. This gated mechanism allows LSTMs to selectively retain or discard information over extended sequences, granting them insensitivity to gap length and making them ideal for time-series prediction tasks.

Each LSTM unit consists of a cell state c_t that acts as a long-term memory storage, and a hidden state h_t that represents the network's output at each time step. The gates—forget gate f_t , input gate i_t , and output gate o_t —modulate the flow of information using sigmoid activations to determine retention values between 0 and 1.

The forget gate decides which information from the previous cell state c_{t-1} to discard:

$$f_t = \sigma\big(W_f \cdot [h_{t-1}, x_t] + b_f\big)\#(6)$$

Where:

- σ is the activation function (e.g., sigmoid),
- Wf is the weight matrix for the forget gate,
- $[h_{t-1}, xt]$ is the concatenation of the previous hidden state and the current input,
- bf is the bias term for the forget gate.

The input gate regulates the new information to be added to the current cell state:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)\#(7)$$

The candidate memory cell update is given by:

$$\widehat{C_t} = tanh(W_c \cdot [h_{t-1}, x_t] + b_c)\,\#(8)$$

The updated memory cell state is computed as:

$$C_t = f_t \odot C_{t-1} + i_t \odot \widehat{C_t}\#(9)$$

where:

- ⊙ denotes element-wise multiplication,
- tanh is the hyperbolic tangent activation function.

Finally, the output gate controls which part of the updated cell state is outputted:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)\#(10)$$
$$h_t = o_t \odot tanh(c_t)\,\#(11)$$

This iterative block structure enables LSTMs to maintain temporal coherence across sequences, making them highly effective in modelling dynamic system behaviours such as DC motor speed prediction under varying conditions [13]. Their predictive accuracy in nonlinear control systems underscores their relevance in intelligent control engineering applications.

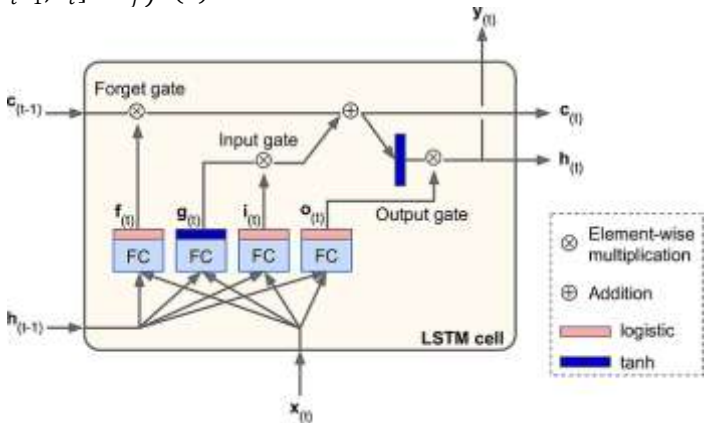

Fig 3. LSTM Cell

V. METHODOLOGY

A. DC Motor Model Assumptions

While the DC motor model presented here is comprehensive, certain assumptions are made to simplify the analysis:

- Linearity: The relationships between voltage, current, torque, and speed are assumed to be linear. Nonlinear effects such as magnetic saturation are neglected.
- Constant Parameters: Parameters such as $(R_a), (L_a), (K_e), (K_t), (J), and (b)$are assumed to remain constant. Variations due to temperature, wear, or other factors are ignored.
- Ideal Conditions: Effects such as eddy currents, hysteresis, and bearing friction are not explicitly modeled.

These assumptions ensure that the model remains tractable for simulation and control design

International Journal of Advances in Engineering and Management (IJAEM)
Volume 7, Issue 06 June 2025, pp: 360-368 www.ijaem.net ISSN: 2395-5252

while still capturing the essential dynamics of the motor.

B. DC Motor Model Parameters

To ensure the simulation reflects real-world conditions, we define the parameters of a 0.5 HP, 6000 RPM DC motor. These parameters are chosen based on typical values for small industrial motors, which are commonly used in robotics, automation, and precision control applications. The selected parameters are as follows:

- Armature Resistance (Ra): 0.027 Ω
- Armature Inductance (La): 0.002 H
- Back EMF Constant (Ke): 0.0382 V·s/rad
- Torque Constant (Kt): 0.0382 N·m/A
- Moment of Inertia (J): 5×10^{-5} kg·m²
- Viscous Friction Coefficient (b): 5×10^{-3} N·m·s/rad

C. Simulating PID-Controlled DC Motor

To simulate the behavior of a PID-controlled DC motor, the mathematical model of the motor described in Section III is integrated with the PID control law. The simulation involves numerically solving the coupled electrical and mechanical equations while dynamically updating the control signal based on the error, integral term, and derivative term. Below is step by step instruction on the simulation is achieved using python programming language.

D. Key Steps in DC Motor Model Simulation

1. Initialization:
- Define initial conditions, including initial speed ω_0, set speed ω_{set}, load torque T_L, and PID gains K_p K_i K_d.
- Specify simulation parameters time step = $1*10^{-4}$ seconds and maximum simulation time of 1.5 seconds.

2. Dynamic Calculation of Error Terms:
- At each time step, compute the error e(t), integral term $\int e(t)$, dt, and derivative term $\frac{de(t)}{dt}$ using numerical methods.

3. Control Signal Computation:
- Apply the PID control law to calculate the applied voltage V(t).

4. Motor Dynamics Simulation:
- Update the motor's speed and current using the electrical and mechanical equations:

$$V(t) = R_a \cdot I(t) + L_a \cdot \frac{dI(t)}{dt} + E_b(t) \#(12)$$

$$J \cdot \frac{d\omega(t)}{dt} = T_m(t) - T_L(t) - b \cdot \omega(t) \#(13)$$

Where Eq. (7) and Eq. (8) are electrical and mechanical equations consecutively.

5. Performance Evaluation:
- Analyze the motor's response by calculating key performance metrics, as described in the next section.

E. Performance Metrics

The performance of the PID-controlled DC motor is evaluated using several key metrics derived from the step response of the system. These metrics provide quantitative insights into the controller's ability to regulate the motor's speed effectively. The equations for each metric are as follows:

1. Rise Time (tr): The time taken for the motor speed to reach 90% of the set speed.
2. Settling Time (ts):The time required for the motor speed to stabilize within ±2% of the set speed.
3. Overshoot (Mp): The maximum percentage deviation above the set speed during the transient phase.

$$M_p = \frac{max(\omega(t)) - \omega_{\text{set}}}{\omega_{\text{set}}} \cdot 100\backslash\%\#(14)$$

4. Steady-State Error (ess): The difference between the final speed and the set speed after the transient phase.

$$e_{ss} = \lim_{t\to\infty} |\omega(t) - \omega_{\text{set}}| \#(15)$$

5. Peak Time (tp): The time at which the motor speed reaches its first peak.
6. Damping Ratio (ζ): A measure of how oscillatory the system is. It is derived from the second-order system approximation.

$$\zeta = \frac{-ln(M_p/100)}{\sqrt{\pi^2 + [ln(M_p/100)]^2}} \#(16)$$

7. Natural Frequency (ωn): The frequency at which the system oscillates in the absence of damping.

$$\omega_n = \frac{\pi}{t_p\sqrt{1-\zeta^2}} \#(17)$$

8. Down Time: The duration during which the motor speed falls below the set speed before stabilizing.

$$\text{Down Time} = \int_{t_1}^{t_2} 1\ dt \quad \text{where } \omega(t) < \omega_{\text{set}} \#(18)$$

These metrics are calculated from the simulated data and used to assess the effectiveness of the PID controller under different operating conditions and gain settings.

International Journal of Advances in Engineering and Management (IJAEM)
Volume 7, Issue 06 June 2025, pp: 360-368 www.ijaem.net ISSN: 2395-5252

F. Synthesis of Training Dataset

The synthesis of a high-quality training dataset is a cornerstone in developing machine learning models for controlling DC motors. This dataset serves as the foundation for training LSTM model, enabling it to learn the intricate relationships between dynamic features (e.g., error, integral term, derivative term) and optimal PID parameters $(K_p), (K_i), (K_d)$.

A well-synthesized dataset ensures that the trained models generalize effectively to unseen scenarios, making them robust to variations in operating conditions. For a DC motor control system, the dataset must capture the relationship between:

- Input Features: Error e(t), integral term $\int e(t)\ dt$, and derivative term $\frac{de(t)}{dt}$.
- Output Targets: Optimal PID parameters$(K_p), (K_i), (K_d)$.

By synthesizing the dataset, we can simulate a wide range of realistic operating conditions, ensuring that the models are exposed to diverse scenarios during training. This approach enables the development of adaptive controllers capable of handling dynamic and uncertain environments and to ensure the dataset is representative of real-world applications, operating conditions are randomly sampled within predefined ranges. These conditions include:

1. Initial Speed ω_0:
- Randomly sampled within a range (500 – 3000rpm).
- Represents the motor's starting speed before control is applied.
2. Set speed ω_{set}:
- Randomly sampled within a range (0 – 3000rpm).
- Represents the desired speed that the controller aims to achieve.
3. Load Torque T_L:
- Randomly sampled within a range (0– 0.5 N·m).
- Represents external forces opposing the motor's motion, such as friction or mechanical loads.

These random samples ensure that the dataset captures a variety of scenarios, including low-speed, high-speed, no-load, and loaded conditions. The diversity of these operating conditions enhances the generalization capability of the trained models.

G. Optimization of PID Parameters Using Genetic Algorithm (GA)

For each set of operating conditions, the optimal PID parameters (K_p, K_I, K_D) are determined using a Genetic Algorithm (GA). GA is a heuristic optimization technique inspired by the process of natural selection. It mimics biological evolution by iteratively evolving a population of candidate solutions through processes such as selection, crossover[14], and mutation. In this context, each individual represents a set of PID parameters (K_p, K_I, K_D), and the fitness function evaluates the control performance based on metrics such as rise time, overshoot, and settling time. The process involves the following steps[13]:

1. Encoding of Candidate Solutions:

Each candidate solution represents a set of PID parameters (K_p, K_I, K_D), encoded as a chromosome. For example, a chromosome might be represented as:

$$\text{Chromosome} = [K_p, K_i, K_d] \#(19)$$

2. Initialization of Population:

A population of candidate solutions is initialized randomly within predefined bounds:

$$K_p \in [K_{p,\min}, K_{p,\max}], \quad K_i \in [K_{i,\min}, K_{i,\max}], \quad K_d \in [K_{d,\min}, K_{d,\max}] \#(20)$$

In this context the bound are chosen between (0, 10) for (K_p, K_I, K_D).

3. Fitness Function:

The fitness of each candidate solution is evaluated using an objective function that minimizes a weighted combination of performance metrics. Common metrics include rise time, settling time, overshoot, and steady-state error. For example:

$$\text{Objective Function} = w_1 \cdot \text{Rise Time} + w_2 \cdot \text{Settling Time} + w_3 \cdot |\text{Overshoot}| + w_4 \cdot |\text{Steady-State Error}| \#(21)$$

Where $w_1, w_2, w_3, w_4$are weighting factors that prioritize specific metrics.

4. Selection:

Individuals with higher fitness scores (better performance) are selected for reproduction. Techniques such as roulette-wheel selection or tournament selection are commonly used.

5. Crossover:

Pairs of selected individuals exchange genetic material to produce offspring. For example, single-point or two-point crossover can be applied to combine portions of two parent chromosomes.

International Journal of Advances in Engineering and Management (IJAEM)
Volume 7, Issue 06 June 2025, pp: 360-368 www.ijaem.net ISSN: 2395-5252

6. Mutation:

Random changes are introduced into the offspring to maintain genetic diversity and prevent premature convergence. Mutation ensures that the algorithm explores new regions of the search space.

7. Termination Criteria:

The algorithm terminates when a predefined number of generations is reached or when the fitness score converges to an acceptable level.

The output of the GA is the set of optimal PID parameters (K_p, K_I, K_D), for each set of operating conditions. These parameters are then paired with the corresponding dynamic features to form input-output pairs.

H. Construction of the Dataset

The synthesized dataset is constructed by pairing the extracted dynamic features (error, integral term, derivative term) with the optimized PID parameters (K_p, K_I, K_D),Since these features are time-series data, they are repeated for each time step during the simulation. The dataset is structured as follows:

1. Inputs (Features):
- Error e(t)
- Integral Term($\int e(t)\ dt$)
- Derivative Term($\frac{de(t)}{dt}$)
2. Outputs (Targets):
- Optimal PID parameters(K_p, K_I, K_D).

The dataset is normalized to improve the performance of machine learning models using standardization technique that scales features to have zero mean and unit variance as follows:

$$x_{\text{normalized}} = \frac{x - \mu}{\sigma} \#(22)$$

Where$(\mu)\ and\ (\sigma)$are the mean and standard deviation of the feature.

Finally, the dataset is split into training and testing sets to evaluate the model's generalization performance. A common split ratio is 80% for training and 20% for testing.

I. LSTM Model Design

The LSTM architecture is designed to process time-series data, where each input sequence corresponds to a series of dynamic features. The configuration includes:

1. Input Layer: Accepts sequences of error, integral term, and derivative term.
2. Hidden Layer: Two stacked LSTM layers with 128 and 64 units, respectively.
3. Dense Layer: A fully connected layer with three neurons and softplus activation for outputting (K_p, K_I, K_D).

J. Training Process

The LSTM model is trained using the synthesized dataset described in Section (H). Key aspects of the training process include using Mean Squared Error (MSE) as a Loss function to measure the difference between predicted and actual PID parameters [15], as for the optimizer, Adam optimizer is employed for efficient gradient-based updates [16, 17].

A batch size of 32 is selected to balance computational efficiency and convergence speed and the model is trained for 200 epochs, with early stopping implemented to prevent overfitting.

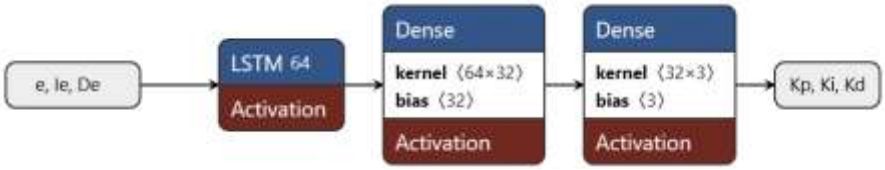

Fig 4. LSTM Architecture

VI. SIMULATION RESULTS

The simulation is based on a high-speed DC motor with parameters carefully selected to represent a small industrial motor suitable for dynamic applications. The key specifications of the motor include an armature resistance$R_a\ of\ 0.027\ \Omega$, armature inductance$L_a\ of\ 0.002\ H$, back EMF constantK_e and torque constant(K_t)of $0.0382\ \text{V} \cdot \text{s/rad}$, and $0.0382\ \text{N} \cdot \text{m/A}$ respectively, a moment of inertiaJ of $5 \times 10^{-5}\ \text{kg} \cdot \text{m}^2$, and a viscous friction coefficient (b) of $5 \times 10^{-3}\ \text{N} \cdot \text{m} \cdot \text{s/rad}$. These parameters define the motor's electrical and mechanical behavior during operation, The simulation time stepdt was set to 1×10^{-4} s, with a maximum simulation duration of1.5sPID parameters were updated at regular intervals0.01sto ensure real-time adaptability. Assess the robustness and adaptability of the control strategies, three distinct trailer scenarios were simulated, each with varying target speeds,

International Journal of Advances in Engineering and Management (IJAEM)
Volume 7, Issue 06 June 2025, pp: 360-368 www.ijaem.net ISSN: 2395-5252

initial speeds, and load torques. The results are presented in tabular form for each trial, followed by step response figures that illustrate the performance of LSTM model.

1. Trail 1 (Low Speed Operation): initial speed is set to 500 RPM, Target Speed of 1500 RPM and Load Torque of 0.2 N.m.
2. Trail 2 (High Speed Operation): initial speed is set to 1000 RPM, Target Speed of 2000 RPM and Load Torque of 0.05 N.m.
3. Trail 3 (Variable Speed Operation): initial speed is set to 500 RPM, Target Speed start from 1500 RPM with periodic increment of 500 RPM until the set speed reaches 2500 RPM for every 1 second, then decreases speed by 400 RPM for every 1 second also, until it reaches a set speedof 1700 RPM, the entire simulation duration is 5 seconds and finally a variable load torque which is randomly set with every change in set speed between 0.05 and 0.5 N.m.

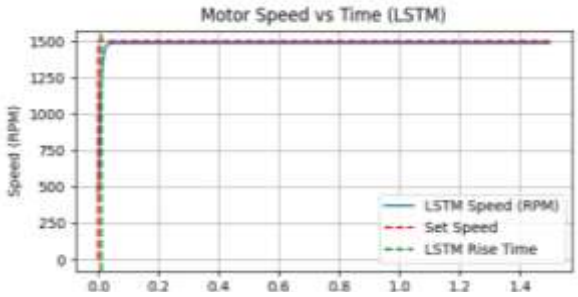

Fig 5. Step Response of LSTM model for Trial 1 (Low-Speed Operation)

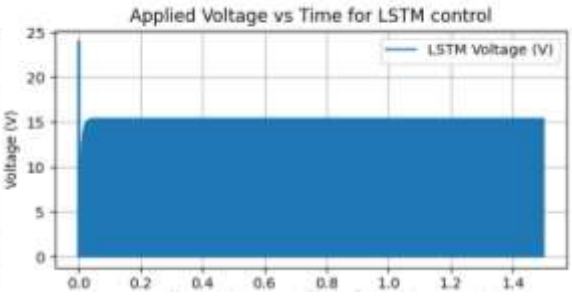

Fig 6. Voltage output of LSTM model for trail 1

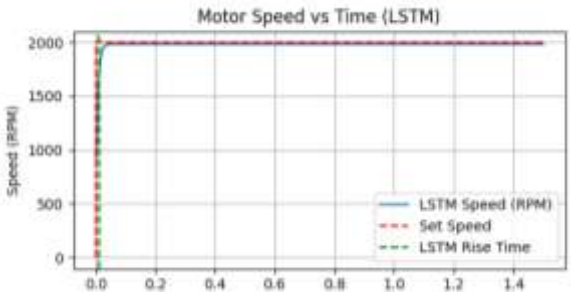

Fig 7. Step Response of LSTM model for Trial 2 (High-Speed Operation)

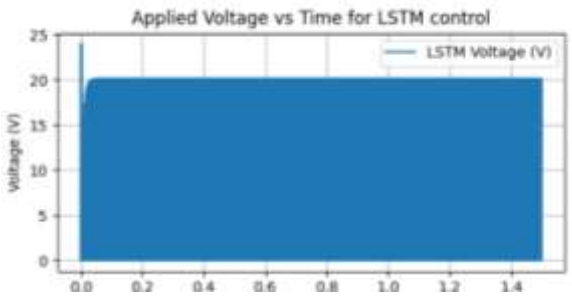

Fig 8. Voltage output of LSTM model for trail 2

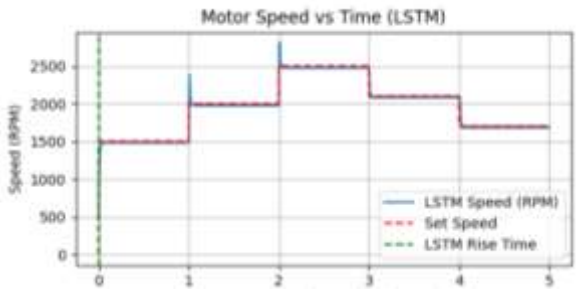

Fig 9. Step Response of LSTM model for Trial 3 (Variable-Speed

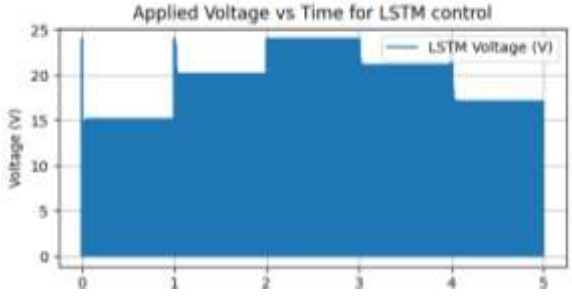

Fig 10. Voltage output of LSTM model for trail 3

International Journal of Advances in Engineering and Management (IJAEM)
Volume 7, Issue 06 June 2025, pp: 360-368 www.ijaem.net ISSN: 2395-5252

Variable-Torque Operation)

Table 1. Performance metrics for Trail 1 to 3

LSTMMetric	Trail 1	Trail 2	Trail 1
Rise Time (s):	0.0121	0.0102	0.0053
Settling Time (s):	0.0294	0.0281	0.0152
Down Time (s):	inf	inf	0.0278
Overshoot (%):	-0.7775	-0.7642	5.8948
Steady-State Error:	12.243	15.7683	15.2244
Peak Time (s):	0.2461	0.2724	0.1493
Damping Ratio:	0.8397	0.8405	0.7063
Natural Frequency (rad/s):	23.5038	21.2879	190.6406

VII. CONCLUSION

This paper demonstrated that Long Short-Term Memory (LSTM) networks excel in DC motor speed control, consistently achieving the shortest rise and settling times. While showing some overshoot, LSTM's strength lies in capturing temporal dependencies for complex, high-speed dynamics, ensuring smooth and stable control. This advanced control strategy holds significant potential for dynamic environments like autonomous vehicles and industrial automation due to its superior generalization capabilities, albeit requiring higher computational resources for this enhanced adaptability. The paper underscores LSTM's substantial contribution as an advanced control strategy, proving its excellence in rapid response and robustness in complex, time-dependent scenarios, despite the inherent trade-off in computational demands.

REFERENCIES

[1]. K. J. Åström and B. Wittenmark, Adaptive Control, 2nd ed. Reading, MA: Addison-Wesley, 1995, pp. 1-20.

[2]. N. Mohan, T. M. Undeland, and W. P. Robbins, Power Electronics: Converters, Applications, and Design, 3rd ed. Hoboken, NJ: Wiley, 2003, ch. 8.

[3]. P. C. Sen, Principles of Electric Machines and Power Electronics, 3rd ed. Hoboken, NJ: Wiley, 2013, pp. 45-60.

[4]. J. J. E. Slotine and W. Li, Applied Nonlinear Control. Englewood Cliffs, NJ: Prentice Hall, 1991, pp. 123-145.

[5]. S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Comput., vol. 9, no. 8, pp. 1735-1780, Nov. 1997.

[6]. B. K. Bose, "Neural network applications in power electronics and motor drives—An introduction and perspective," IEEE Trans. Ind. Electron., vol. 54, no. 1, pp. 14-33, Feb. 2007.

[7]. Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, no. 7553, pp. 436-444, May 2015.

[8]. M. Azri and B. A. Mutalib, "Speed Control of Dc Motor Using Pi Controller Mohd Azri bin Abd Mutalib," 2008. Accessed: Mar. 15, 2025. [Online]. Available: https://www.semanticscholar.org/paper/Speed-Control-of-Dc-Motor-Using-Pi-Controller-Mohd-Azri-Mutalib/223931aa76bdd865a54ae28ccc8f54af28c5ce8c

[9]. Tan Kiong Howe, "Evaluation of the Transient Response of a DC motor using MATLAB/SIMULINK," May 2003.

[10]. Muhammed Nasiruddin Bin Mahyddin, "Direct model reference adaptive control of coupled tank liquid level control system," Nov. 2005.

[11]. K. J. Åström, T. Hägglund, and K. J. Åström, PID controllers, 2nd ed. Research Triangle Park, N.C: International Society for Measurement and Control, 1995.

[12]. S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Comput., vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[13]. B. K. Bose, "Neural Network Applications in Power Electronics and Motor Drives—An Introduction and Perspective," IEEE Trans. Ind. Electron., vol. 54, no. 1, pp. 14–33, Feb. 2007, doi: 10.1109/TIE.2006.888683.

[14]. T. A. Faris Ku Yusoff, M. F. Atan, N. Abdul Rahman, S. F. Salleh, and N. A.

Wahab, "Optimization of PID Tuning Using Genetic Algorithm," J. Appl. Sci. Process Eng., vol. 2, no. 2, Jan. 1970, doi: 10.33736/jaspe.168.2015.

[15]. I. Goodfellow, Y. Bengio, and A. Courville, Deep learning. in Adaptive computation and machine learning. Cambridge, Massachusetts: The MIT Press, 2016.

[16]. D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," 2014, arXiv. doi: 10.48550/ARXIV.1412.6980.

[17]. D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," 2014, arXiv. doi: 10.48550/ARXIV.1412.6980.