mathematics

MDPI

Article

Artificial Neural Network-Based Data-Driven Parameter Estimation Approach: Applications in PMDC Motors

Faheem Ul Rehman Siddiqi [1,†], Sadiq Ahmad [1], Tallha Akram [2,†], Muhammad Umair Ali [3,†], Amad Zafar [3,*] and Seung Won Lee [4,5,6,7,*]

[1] Department of Electrical Engineering, COMSATS University Islamabad, Wah Cantonment 47040, Pakistan; faheemsiddiqi789@gmail.com (F.U.R.S.)
[2] Department of Information Systems, College of Computer Engineering and Sciences, Prince Sattam Bin Abdulaziz University, Al-Kharj 11942, Saudi Arabia
[3] Department of Artificial Intelligence and Robotics, Sejong University, Seoul 05006, Republic of Korea
[4] Department of Precision Medicine, Sungkyunkwan University School of Medicine, Suwon 16419, Republic of Korea
[5] Department of Metabiohealth, Sungkyunkwan University, Suwon 16419, Republic of Korea
[6] Personalized Cancer Immunotherapy Research Center, Sungkyunkwan University School of Medicine, Suwon 16419, Republic of Korea
[7] Department of Artificial Intelligence, Sungkyunkwan University, Suwon 16419, Republic of Korea
* Correspondence: amad@sejong.ac.kr (A.Z.); swleemd@g.skku.edu (S.W.L.)
† These authors contributed equally to this work.

Abstract: The optimal performance of direct current (DC) motors is intrinsically linked to their mathematical models' precision and their controllers' effectiveness. However, the limited availability of motor characteristic information poses significant challenges to achieving accurate modeling and robust control. This study introduces an approach employing artificial neural networks (ANNs) to estimate critical DC motor parameters by defining practical constraints that simplify the estimation process. A mathematical model was introduced for optimal parameter estimation, and two advanced learning algorithms were proposed to efficiently train the ANN. The performance of the algorithms was thoroughly analyzed using metrics such as the mean squared error, epoch count, and execution time to ensure the reliability of dynamic priority arbitration and data integrity. Dynamic priority arbitration involves automatically assigning tasks in real-time depending on their relevance for smooth operations, whereas data integrity ensures that information remains accurate, consistent, and reliable throughout the entire process. The ANN-based estimator successfully predicts electromechanical and electrical characteristics, such as back-EMF, moment of inertia, viscous friction coefficient, armature inductance, and armature resistance. Compared to conventional methods, which are often resource-intensive and time-consuming, the proposed solution offers superior accuracy, significantly reduced estimation time, and lower computational costs. The simulation results validated the effectiveness of the proposed ANN under diverse real-world operating conditions, making it a powerful tool for enhancing DC motor performance with practical applications in industrial automation and control systems.

Keywords: PMDC; ANN; parameter estimation; feedforward network; Bayesian regularization; Levenberg–Marquardt

MSC: 68T05; 68T07

check for updates

Citation: Siddiqi, F.U.R.; Ahmad, S.; Akram, T.; Ali, M.U.; Zafar, A.; Lee, S.W. Artificial Neural Network-Based Data-Driven Parameter Estimation Approach: Applications in PMDC Motors. *Mathematics* **2024**, *12*, 3407. https://doi.org/10.3390/math12213407

Academic Editors: Helga Silaghi and Claudiu Raul Costea

Received: 30 September 2024
Revised: 26 October 2024
Accepted: 29 October 2024
Published: 31 October 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

A power system is a complex network that delivers electrical energy from generating plants to consumers at the distribution level via transmission lines. On the distribution side, consumers predominantly use direct current (DC) electrical drives, which are crucial in their daily lives, and their use is steadily increasing. DC electrical drives provide advantages

such as high efficiency, wide-speed control, and rapid response. The main component is the DC motor, which is widely used owing to its simplicity, easy control, and high performance. DC drives are widespread in industries and household appliances. Their ease of control and high efficiency make them the first choice for manufacturers. Permanent-magnet DC (PMDC) motors are simple, easy to use, dependable, and efficient, making them very useful in various industries and daily applications [1].

PMDC motors are used in the automobile industry in applications such as power windows, windscreen wipers, and cooling fans. They are common in home equipment, such as electric shavers, hair dryers, blenders, and vacuum cleaners. The list extends to electric tools, such as screwdrivers and drills, which are also used in the medical field for devices such as infusion pumps and other portable medical equipment requiring small, lightweight motors. Due to the widespread use of DC motors, one cannot overlook the aspect of parameter estimation, which is crucial for controlling, maintaining stability, ensuring efficiency, and enhancing the reliability of the motor. It is essential to obtain precise parameters to achieve control responses, develop a system model, design control techniques, and extract mathematical models.

Precise parameter estimation using an accurate model is crucial in such scenarios. Various techniques have been employed to achieve this objective. For example, optimization techniques such as whale optimization have been used for parameter estimation [2]. The recursive least-squares method was employed to estimate the unknown parameters in DC motors [3]. Furthermore, a comparison between the non-linear least-squares method and pattern search optimization algorithms for parameter estimation were presented in reference [4]. Real-time parameter estimation in PMDC motors was performed using quantized sensors, as described in reference [5]. This technique was further analyzed through online parameter estimation of a PMDC motor. The output from the quantized sensors was utilized for online parameter estimation of the DC motor, as discussed in [6]. Here, we provide an extensive example to demonstrate various estimation applications.

A method for estimating the DC motor was proposed by considering the effects of the armature teeth slots and commutation, as outlined in [7]. Various motors were evaluated, and the collected data were analyzed to obtain an optimal solution. One mathematical optimization method for handling parameter estimation issues is multiparametric programming. This approach involves estimating the system parameters by minimizing the cost function dependent on constraints, as discussed in [8]. An innovative method for parameter estimation based on the Rao-1 algorithm and a modified Kalman filter was proposed in [9]. One of the most widely used optimization techniques is a genetic algorithm (GA) [10], which can easily perform parameter estimation. In reference [11], GA-based grey wolf optimization (GWO) was used. However, because the algorithm involved a considerable amount of data, more time was required to obtain the estimated parameters for the brushed DC motor.

Several approaches have been employed for measuring the parameters of a DC motor [12], and these approaches are related to machine learning to a certain extent. In reference [13], only the temperature was estimated using different machine learning algorithms such as regression, a support vector machine (SVM), and decision trees. Similarly, in reference [14], a regression model was utilized to identify a DC motor; however, parameter estimation was not performed. A cascade feedforward neural network (CFNN) was utilized to simultaneously estimate the parameters and state of a DC motor [15]. The rotor resistance, temperature, and speed of the motor were estimated using this technique in addition to Bayesian regularization backpropagation (BRBP). Power consumption for a DC motor was estimated by [16] using artificial neural networks (ANNs). Two-dimensional modeling of the external rotor brushless DC (BLDC) motor using finite element analysis (FEA) software was employed. The motor's power was measured using ANNs, SVMs, and extreme learning machines (ELMs) to obtain estimated motor power values. Several recent studies have also demonstrated progress in estimation using machine learning algorithms, but these can only measure one or two parameters [17]. For instance, the Kalman Filter

algorithm improves the accuracy of DC motor speed estimation by reducing the noise ripple and optimizing the filter parameters to improve performance and stability. However, only one parameter, i.e., speed, was estimated [18]. A fractional-order adaptive Kalman filter (FOAKF) was used for sensorless DC motor speed estimation, which is superior to the previous method [19]. Similarly, in Ref. [20], the position and speed of BLDC were estimated using dual ANNs with high accuracy. FEA and genetic programming (GP) have been used to predict the torque of brushless DC motors [21]. Parametric simulations and bidirectional LSTM AI models were used to estimate the rotor speed of the BLDC motors [22]. A hybrid estimator combining Machine Learning (ML) and Extended Kalman Filter (EKF) techniques accurately estimates BLDC motor speed and back-EMFs, with an Artificial Neural Network (ANN) showing the highest accuracy [23]. Recently, a comparative study of the Steiglitz–McBride, Jaya, GA, and GWO algorithms for parameter estimation of DC motor dynamic models revealed GWO's high accuracy but at a higher computational cost, whereas Steiglitz–McBride offers a better balance between accuracy and efficiency [24]. All these techniques estimate only one or two parameters, but our model can estimate all the parameters for better accuracy and control of PMDC motors.

This study presented a novel approach for estimating the critical parameters of a PMDC motor using an ANN. Furthermore, it addresses the challenge of limited motor-characteristic information by defining practical constraints to simplify the estimation process. A mathematical model for optimal parameter estimation was developed, and two advanced learning algorithms were introduced to train the ANN efficiently. The ANN-based estimator predicts the motor's key electromechanical and electrical characteristics, such as the back-EMF, moment of inertia, viscous friction coefficient, armature inductance, and armature resistance.

2. Modeling of a PMDC Motor

The design of a permanent-magnet DC motor comprises two main parts. One is the electrical part, and the other is the mechanical part of the motor. The electrical component contains the armature resistance (R_a) and armature inductance (L_a). The mechanical parts have a moment of inertia (J_{eq}) and a viscous friction constant (B). When the motor is in motion, it produces torque due to its angular speed (ω_r). When the motor is started, it produces a back-EMF voltage, which is shown as V_b in Figure 1.

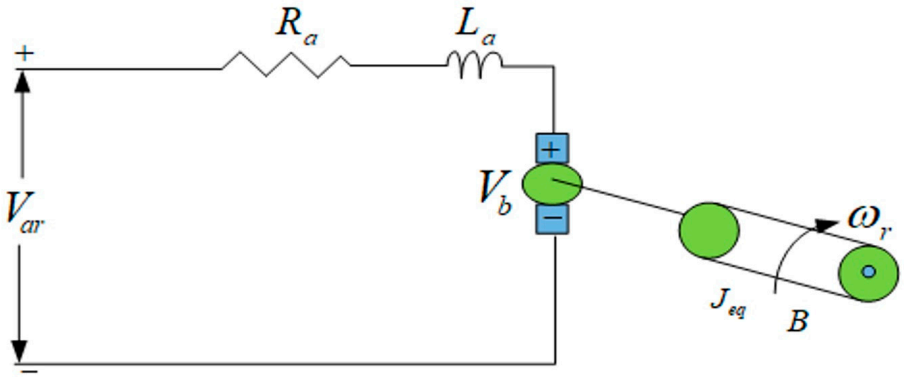


Figure 1. PMDC Model Diagram [1].

The PMDC model parameters utilized are summarized in Table 1, which is taken from an openly available dataset [25]. These are the actual values used to compare the estimated and actual values of the PMDC motor parameters. The equation for the PMDC motor shown in Figure 1 is as follows:

$$V_{ar} = i_a R_a - L_a \frac{di_a}{dt} - V_b \tag{1}$$

Table 1. Actual available data on a PMDC motor [25].

Parameters	Values
Armature Resistance (R_a)	2.0 Ω
Armature Inductance (L_a)	0.5 H
Moment of Inertia (J_{eq})	0.02 Kgm/s^2
Viscous Friction Coefficient (B)	0.2 N.m.sec
Back-Electromotive Force (EMF) Constant K_b	0.1 V/rad/sec

3. Problem Formulation

The PMDC motor model was transformed with error minimization of the given mathematical model. A cost function for minimization was created using the sum of squared errors (SSE). The difference between the actual speed (i.e., the output speed of the motor y_{act}) and the output speed of the estimated model (y_{est}) is the error signal. Different optimization methods, genetic algorithms, and particle swarm optimization are efficient; however, all these methods are stuck at a certain level and are unable to match the actual values of the parameters.

To improve the precision of the estimated parameters, namely the armature inductance, armature resistance, torque constant, viscous friction constant, and moment of inertia, a machine learning algorithm was used to address this issue. The error between the calculated speeds was minimized by optimizing the estimated parameters, resulting in a more accurate representation of the dynamic behavior of the DC machine. The complexity of the problem increases with the absolute error owing to the disparity between the measured and estimated data. The problem was solved by applying a machine learning algorithm to achieve the minimum difference between the measured and estimated values and to obtain an optimal solution with maximum efficiency and correct parameter estimation in less time.

This study has two sub-frameworks: the ML algorithm and a comparison with the optimization technique. A dataset was provided to the ML algorithm to predict the estimated values of the parameters, and then a comparison was made with a previously used optimization technique.

$$y_{act} = \frac{K_b}{(J_{eq}s + B)(L_a s + R_a) + K_b^2} \tag{2}$$

$$y_{est} = \frac{K_{dc}\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{3}$$

In Equation (2), K_b represents the back-EMF constant. In Equation (3), K_{dc}, ζ, and ω_n will be calculated by

$$K_{dc} = \frac{1}{BR_a + K_b^2} \tag{4}$$

$$\zeta = \frac{J_{eq}R_a + BL_a}{2\sqrt{(BR_a + K_b^2)(J_{eq}L_a)}} \tag{5}$$

$$\omega_n = \sqrt{\frac{BR_a + K_b^2}{J_{eq}L_a}} \tag{6}$$

4. Artificial Neural Networks

Artificial Neural Networks (ANNs) are computational models that emulate neural structures to facilitate pattern recognition, data classification, and predictive analytics across various applications, including machine learning, natural language processing, and image recognition [26]. With advancements in deep learning, ANNs, particularly deep neural

networks, have evolved to handle increasingly complex tasks, contributing significantly to AI progress [27].

A neural network is defined as a collection of interconnected artificial neurons, also referred to as units, arranged in a layer-by-layer configuration [28]. Fundamentally, an ANN comprises three primary layers: input, hidden, and output layers. The weighted directed graph is mostly used to view an ANN, with artificial neurons acting as nodes. The input signal for the ANN can be a vector picture and pattern from an external source [29]. Further details related to an ANN's functionality can be found in the literature [26–30]. In cases where there is insufficient data to train the model, such as rare or extreme events, ANN application is not feasible [30].

5. The ANN's Algorithm

5.1. Multilayer Perceptron Neural Network

A feedforward neural network with a hidden layer positioned between the input and output layers is called a multilayer perceptron (MLP). The optimal option is to use a single hidden layer to render the MLPs a global approximator. A crucial step is to choose the hidden layers to include the number of neurons in each hidden layer. Increasing both variables improves the learning capacity, but the network responds slowly. This selection procedure is often empirical and issue-specific.

$$y_s = \delta_0 \left(\sum_{j=0}^{N} \omega_{sj}^{y} \left(\delta_H \left(\sum_{i=0}^{M} \omega_{ji}^{H} x^i \right) \right) \right) \tag{7}$$

- ω_{sj}^{y} symbolizes the weights that are transferred from the hidden layer's neuron j to the k^{th} output neurons.
- ω_{ji}^{H} portrays the weights that are transferred from neuron i in the input layer to neuron j in the hidden layer.
- x^i corresponds to the i^{th} element in the input layer.
- The activation functions in the hidden and output layers are denoted as δ_H and δ_0.

It is necessary to minimize the following cost function:

$$\min_{J_{eq,B,\ La,Ra,K_b}} \qquad \sum_{k=1}^{N} |y_{act}(k) - y_{est}(k)|^2 \tag{8}$$

5.2. Bayesian Backpropagation

To enhance the generalizability of artificial neural network models for accurate forecasting, the training datasets should be embedded within the Bayesian paradigm. It is a well-organized and all-embracing structure created by Bayesian probability theory, which enables it to address uncertainty with simplicity.

Bayesian Regularization Framework

To optimize regularization parameters using Bayesian theory, the minimal point P_{min} of a given function $G(\omega)$ must be calculated using the Hessian matrix. The Levenberg–Marquardt optimization approach [31], which is based on the Gauss–Newton approximation, was used to determine the minima.

$$\Delta\omega_i = -\left[\nabla^T(\omega_i)\nabla(\omega_i) + \zeta_i I\right]^{-1} \nabla^T(\omega_i) V(\omega_i) \tag{9}$$

where ζ is the step-regulating parameter and $\nabla(\omega_i)$ is the Jacobian matrix. The sum of squares function is denoted by $V(\omega_i)$.

$$G(\omega) = \sum_{k=1}^{N} V_k^2(\omega) = V^T(\omega) V(\omega) \tag{10}$$

The following relations are used to determine the regularization parameters G and K:

$$\Phi_{min} = \frac{\zeta}{2\gamma_w W_{min}} \tag{11}$$

$$\Psi_{min} = \frac{N-\zeta}{2\gamma_D W_{min}} \tag{12}$$

Here, the effective number of parameters utilized to lower the error function is shown by ζ. The definition of the variable ζ is given by:

$$\zeta = n - 2\Phi_{min} tr(H_{min})^{-1} \tag{13}$$

$H = 2\Psi\nabla^T\nabla + 2\Phi\ I_n$ may be used to construct the Hessian matrix, where n is the overall number of network parameters. The objective function is defined as

$$G(\omega) = \Psi\gamma_D + \Phi\gamma_E \tag{14}$$

Alternatively, the function can be expressed in terms of the mean square term:

$$G(\omega) = \Psi\sum_{m}^{M}\left(\Gamma_{tar} - \Gamma_{act}^T\right)(\Gamma_{tar} - \Gamma_{act}) + \phi\sum_{j}^{n}\omega_j^2 \tag{15}$$

where Γ_{act} denotes the vector representing the provided network's actual response and Γ_{tar} is the target vector. The pseudo-code for the Bayesian backpropagation is described in Algorithm 1.

Algorithm 1 Bayesian Backpropagation

Result: Weight & biases update for curve approximation
Initialization: Initialize Φ, ψ, and weights vector randomly **while** $TolLevel < \epsilon$ **do**

- Compute γ_D and γ_E: First, calculate ϕ and ψ by setting $\zeta = n$ using Equations (5) and (6).
- Take the initial step toward minimizing the cost function in Equation (8) using the Levenberg-Marquardt algorithm.
- Compute ζ using Equation (7). The Hessian Matrix H is calculated using relation $H = 2\Psi\nabla^T\nabla + 2\Phi\ I_n$

end

5.3. Levenberg–Marquardt Algorithm

A new version of Newton's technique for minimizing functions that is the sum of the squares of other non-linear functions is the Levenberg–Marquardt algorithm. This is effective for neural network training, using the MSE as the performance metric.

$$x_{n+1} = x_n - A_n^{-1} g_n \tag{16}$$

where $A_n = \nabla^2 G(x)\big|_{x=x_n}$ and $g_n = \nabla G(x)|_{x=x_n}$. If we assume that it is a sum of squares function:

$$G(x) = \sum_{j}^{M} v_j^2(x) = v^T(x)v(x) \tag{17}$$

then the n^{th} element of the gradient would be

$$[\nabla G(x)]_n = \frac{\partial G(x)}{\partial x_n} = 2\sum_{j=1}^{M} v_j(x)\frac{\partial v_j(x)}{\partial x_n} \tag{18}$$

The gradient can, therefore, be written in matrix form:

$$\nabla\, G(x) = 2K^T(x)v(x) \tag{19}$$

where

$$K(x) = \begin{bmatrix} \frac{\partial v_1(x)}{dx_1} & \frac{\partial v_1(x)}{dx_2} & \cdots & \frac{\partial v_1(x)}{dx_n} \\ \frac{\partial v_2(x)}{dx_1} & \frac{\partial v_2(x)}{dx_2} & \cdots & \frac{\partial v_2(x)}{dx_n} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial v_1(x)}{dx_1} & \frac{\partial v_1(x)}{dx_2} & \cdots & \frac{\partial v_M(x)}{dx_n} \end{bmatrix} \tag{20}$$

Next, we determine the Hessian matrix. The elements of the Hessian matrix are

$$\left[\nabla^2 G(x)\right]_{n,j} = \frac{\partial^2 G(x)}{\partial\, x_n \partial\, x_j} = 2\sum_{j=1}^{M}\left\{\frac{\partial v_j(x)}{\partial\, x_n}\frac{\partial\, v_j(x)}{\partial\, x_j} + \partial\, v_j(x)\,\frac{\partial^2\, v_j(x)}{\partial\, x_n \partial\, x_j}\right\} \tag{21}$$

The Hessian matrix can then be expressed in matrix form:

$$\nabla^2 G(x) = 2K^T(x)K(x) + 2S(x) \tag{22}$$

where

$$S(x) = \sum_{j=1}^{M} v_j(x)\nabla^2 v_j(x) \tag{23}$$

Assuming $S(x)$ is small, the following approximation can be used to obtain the Hessian matrix:

$$\nabla^2 G(x) = 2K^T(x)K(x) \tag{24}$$

The pseudo-code for the Levenberg–Marquardt algorithm is described in Algorithm 2.

Algorithm 2 Levenberg–Marquardt algorithm

Step 1. Initialize: Initialize neural network weights θ_0 and Levenberg–Marquardt algorithm parameters: λ (damping parameter), ϵ (tolerance) randomly.
Step 2. Forward pass: Compute the output of the neural network for all input samples, which can be evaluated by $\hat{y} = f\left(\theta_j,\ X\right)$
Step 3. Backward pass: Compute the error between the network output and the target output $E = Y - y$, and compute the gradient of the error with respect to the network weights : $\Delta E = \frac{E}{\theta_j}$.
Step 4. Jacobian matrix: Compute the partial derivatives of the network output with respect to each weight parameter where $K = \frac{\partial\, f\left(\theta_j, X\right)}{\partial \theta_j}$
Step 5. Hessian matrix: Compute the outer product of the Jacobian matrix with itself by computing $H = K^T\, K$
Step 6. Damped least squares approximation: Compute the damped least squares approximation of the Hessian matrix by using $H^\lambda = H + \lambda I$.
Step 7. Parameter update: Solve the linear system for the parameter update can be evaluated by $\Delta\theta_j = \left(H^\lambda\right)^{-1}\nabla E$
Step 8. Update network weights: Update each weight using the corresponding parameter update with the help of $\theta_{j+1} = \theta_j + \Delta\theta_j$
Step 9. Evaluate change in error: If $\left|\left|E_{j+1}\right|\right| < \left|\left|E_j\right|\right|$ then decrease λ otherwise increase λ.
Step 10. Check for convergence: If $\left|\left|E_{j+1} - E_j\right|\right| < \epsilon$, converge.
Step 11. End: Return the trained neural network with weights θ_{j+1}.

6. Results and Discussion

Using machine learning for parameter estimation of DC motors is a relatively new area of exploration. In this study, parameter estimation for a PMDC motor was conducted using an ANN. The principal rationale for adopting ANNs is their ability to effectively

define non-linear relationships between inputs and outputs, particularly for resolving non-linear interdependencies. Moreover, ANNs are scalable and can manage large datasets, making them suitable for applications in which substantial amounts of data are received from sensors and motor controllers. Moreover, ANNs are robust against noisy data and fluctuations in sensor readings, which are common in embedded motor control environments. The cost function is selected as the MSE, where the five parameters of the PMDC motor—armature resistance, armature inductance, moment of inertia, back-EMF constant, and viscous friction constant—are condensed into three parameters to simplify the problem. The new estimated parameters are K_{dc} (DC gain), ω_n (the natural frequency), and ζ (the damping ratio) of the PMDC motor.

The data used to train the neural network were sourced from a publicly available dataset on Kaggle titled "Operation of DC motor" [25]. The dataset was collected by varying the input voltage and torque parameters. The simulated engine parameters are an armature resistance of 2 Ω, an armature inductance of 0.5 H, a friction constant of 0.2 N/rad/s, and a moment of inertia of value 0.02 Kg/m^2. The input voltage varies from 1 to 100, while the torque is input voltage times 0.05.

The dataset contains 18 columns with dimensions of 5150 × 18. It contains voltages, speed at different voltage levels, the electric conversion constant, and the electromechanical conversion constant. Thus, the machine learning algorithm used 18 features and 5150 observations. Eighty percent of the dataset was used for training, while the remaining 20 percent was used to test the ANN model. Because there is no standard procedure to determine the number of hidden layers and neurons in each layer, different configurations are tested. The chosen architecture has two hidden layers, and the number of neurons in each hidden layer is 20 and 10. Hyperparameter tuning was performed by varying the number of epochs, learning rates, activation functions, and weight initializations.

The graph depicts the performance profile of the ANN model, represented by the MSE on the y-axis and epochs on the x-axis, as shown in Figure 2. As the training advances, the graph shows a decreasing MSE, indicating an improvement in the predicted accuracy of the model. Initially, the MSE was high, indicating that the model continually learned and modified its weights. By the 35^{th} epoch, the ANN converged and achieved the optimum performance with a significantly low MSE of 0.0033919. This convergence point shows that the model successfully captured fundamental information patterns while minimizing prediction errors. The consistent decrease in the MSE across epochs highlights the learning process of the model, which is driven by weight optimization via backpropagation and gradient descent. Each epoch represented one complete trip through the training dataset, enabling the ANN to adjust its parameters repeatedly. The low final MSE value demonstrates the model's performance in minimizing the difference between the predicted and actual values, demonstrating its robustness and dependability. The graph shows that the system did not overfit when performing at the lowest MSE. In addition, the system achieved the best value for the testing and training data, with the lowest number of iterations.

The training state graph of the ANN model provides a thorough picture of the numerous key parameters that converged at the 35th epoch, as shown in Figure 3, which consists of five subgraphs: gradient, MU, number of parameters, sum-squared parameters, and validation checks. At convergence, the gradient achieved a minimum of 0.0043, indicating that the optimization procedure successfully fine-tuned the weights. The momentum term (MU) demonstrates the impact of the adaptive learning rate on the training dynamics. The total number of parameters, 605, demonstrates the model's intricacy and ability to understand deep data patterns. The sum-squared parameters, 5.0749×10^{32}, represent the cumulative weight magnitudes, indicating the vast parameter space of the model.



Figure 2. Performance of the ANN.

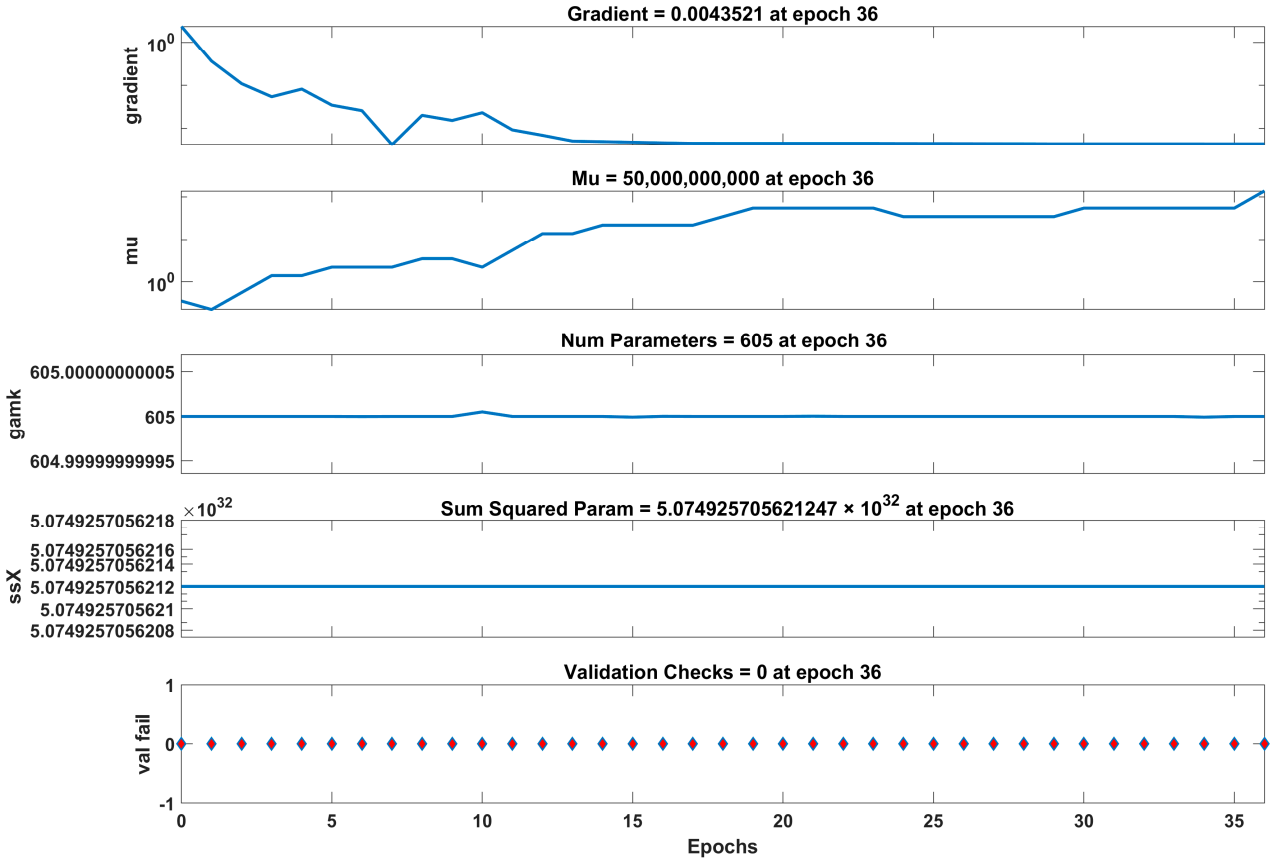


Figure 3. Training State.

Finally, a validation check score of zero indicated no validation failures, ensuring solid efficiency without overfitting. This thorough convergence at the 35th epoch represents the optimal state of the ANN, which balances accuracy with generalization. The convergence of the ANN model at the 35th epoch improved the accuracy and adaptability of the system by successfully optimizing the weights and preventing validation failures. This condition guarantees a balanced model that can make exact predictions while being generalizable, thereby enhancing overall system performance.

After a feedforward neural network is trained, the error histogram is a histogram of the errors between the target and predicted values, as shown in Figure 4. The number of vertical bars in a graph is called a bin. Here, the entire error range was split into 30 smaller bins.

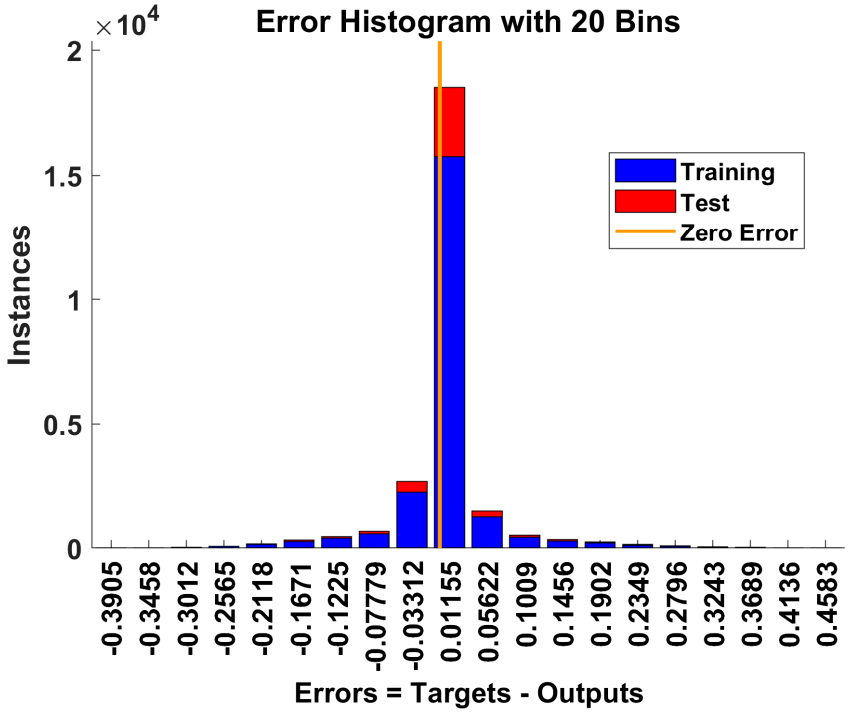


Figure 4. Error Histogram.

The number of samples from the dataset that fall into a certain bin is shown on the y-axis. For instance, a bin in the middle of the plot represents an error of 0.0115. The bin height of the training dataset was below but close to 150, whereas the bin height of the validation and test datasets was between 150 and 200. This indicates that many samples from various datasets have inaccuracies in the following range:

The zero-error value on the error axis (often known as the x-axis) is matched by the zero-error line. As shown in Figure 4, in this instance, a zero-error point is located beneath the bin with a center at 0.0115. The system shows an absolute error histogram, where the error has been minimized to the center, and the system has achieved maximum efficiency in predicting the values.

The different regression results are shown in the following figures, and the test data results are shown in Figure 5. As discussed earlier, the training dataset was 80 percent of the total dataset. This indicates that the model trained the dataset and fitted the neural network with an efficiency of 99.72 percent.

After training the ANN using the training dataset, as discussed earlier, 20 percent of the dataset was used to test the ANN model. This allows us to evaluate the predictive ability of the proposed model. The output of the test dataset is shown in Figure 5. The output shows that the model predicts the test data accurately and efficiently. The model showed a prediction efficiency of approximately 99.69 percent efficiency. This indicates that the training and testing data validated each other. Both were highly efficient. The testing data demonstrated the model's efficiency by showing values that were extremely close to those of the training dataset.

The ANN-based model predicts the same values. Thus, the model has estimated values for the parameters close to their actual values. This indicates that the ANN-based model can estimate the values with an efficiency of 99.71 percent.

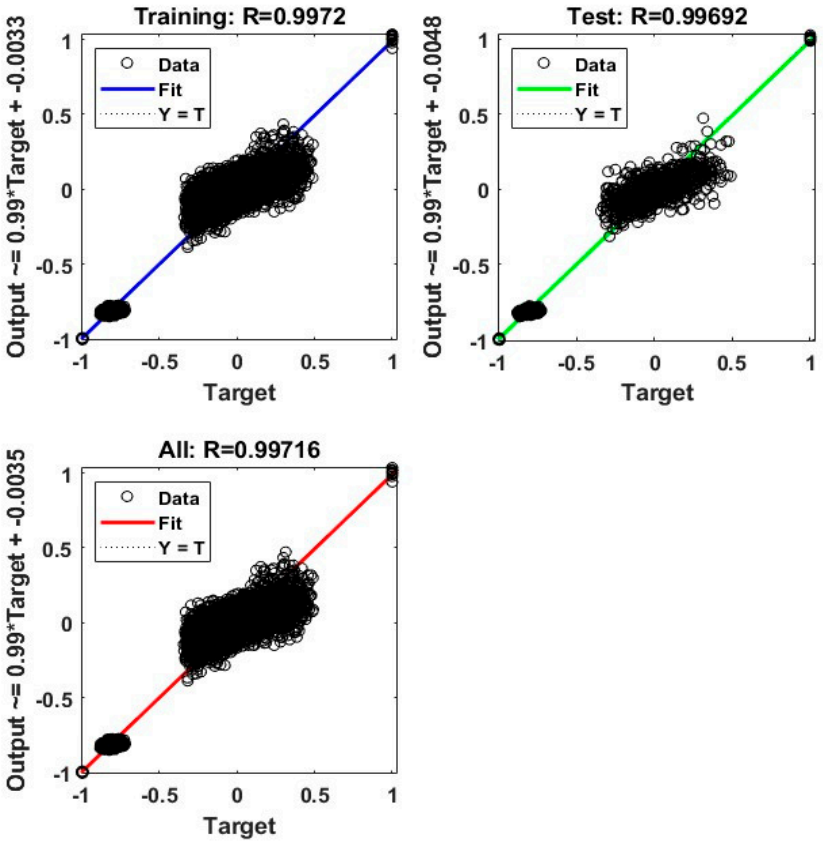


Figure 5. Training Dataset.

The results from implementing the ANN model surpassed those achieved using previously employed optimization techniques for parameter estimation. The ANN model demonstrated remarkable efficiency with an accuracy of up to 99.7 percent. Before this, various optimization techniques were employed for the parameter estimation of DC motors, which yielded satisfactory results to a certain extent. The GA, one of the most effective optimization techniques, produced superior results. The GA exhibited an efficiency of 98.6 percent. Therefore, the ANN demonstrated one percent higher efficiency than the GA. Nevertheless, the ANN parameter estimation outperformed the GA, yielding significantly better results.

A comparison of the estimated and actual parameters (i.e., derived from Equations (4)–(6)) of the PMDC motor is presented in Table 2. This table compares five parameters for a PMDC motor, including real values and estimations from the GA. The DC gain constant (k_{dc}) had an actual value of 0.0999, with ANN and GA estimations of 0.0995 and 0.0993, respectively. Both approaches demonstrated negligible discrepancies and good accuracy, with the ANN being somewhat more accurate. The natural frequency (ω_n) is 4.4743, which nearly matches the ANN estimate of 4.4739, with a small divergence from the GA at 4.4152. The damping ratio (ζ) has an actual value of 1.3409, which is virtually identical to the ANN estimate of 1.3409, whereas the GA estimates 1.3152, showing less accuracy than the ANN. The SSE for both approaches is small, with the ANN at 2.52×10^{-29} and the GA at 1.03×10^{-23}, although the ANN performs better. Finally, the ANN converged in 35 iterations versus the GA in 200, demonstrating the efficacy of the ANN in parameter estimation.

The ANN and GA produced accurate estimates of the PMDC motor's characteristics, although the ANN was more precise and efficient. The ANN technique closely matches the real values for k_{dc}, ω_n, and ζ, with a low SSE error and substantially fewer iterations needed to reach convergence.

Table 2. Comparison of a PMDC Motor's Estimated and Actual Parameters.

Parameters	Actual Values	ANN	GA
k_{dc}	0.099900099	0.09952	0.9937
ω_n	4.474371464	4.47391	4.415215
ζ	1.340970468	1.34097	1.31126
SSEE rror	0	2.52×10^{-29}	1.03×10^{-23}
Iterations	200	35	200

Table 2 shows that the ANN efficiently estimated the parameters of the PMDC motor, outperforming the research employing a GA. Additionally, it is a less time-consuming process with fewer iterations. Previously utilized optimization techniques require numerous iterations and more time to achieve better results. Both machine learning and optimization approaches are highly complex technologies rooted in advanced mathematics. These methods require substantial processing power and data. These technologies profoundly impact current and future decision-making, particularly as data generation advances.

7. Conclusions and Future Work

This study aims to accurately estimate the parameters of a PMDC motor to develop a precise and comprehensive motor model that is critical for optimizing the design and control of DC motors using an artificial neural network (ANN). In this paper, the five parameters of the PMDC motor—armature resistance, armature inductance, moment of inertia, back-EMF constant, and viscous friction constant—have been initially condensed to just three parameters (i.e., dc gain constant, natural frequency, and damping ratio) to simplify the problem. Then, An ANN-based error minimization objective function was formulated, and advanced optimization techniques were applied to achieve accurate parameter estimation. The ANN model has demonstrated remarkable efficiency, estimating values with up to 99.7 percent accuracy. The results demonstrated that the ANN model outperformed traditional convergence rate, accuracy, and computational efficiency methods. Specifically, the ANN achieved rapid convergence with significantly fewer iterations and closer alignment to the true parameter values, leading to lower estimation errors and enhanced model reliability. Comparative analysis indicated that the ANN-based approach yielded more refined and precise parameter estimates, making it a superior PMDC motor modeling and control tool. These findings underscore the effectiveness of ANNs in real-world applications and their potential to drive advancements in motor control strategies.

Future Work

The loading effect on the PMDC motor was overlooked while estimating the DC motor characteristics in this research effort; however, future work can be expanded to examine the impact of loading on the PMDC motor parameters. Hardware with a loading effect can be used to execute the operation, and related parameters can be approximated. In addition, machine learning has numerous algorithms that can be used to estimate the PMDC parameters. Furthermore, other algorithms may be used to produce better and more efficient results.

Author Contributions: Conceptualization, F.U.R.S. and T.A.; Formal analysis: S.A., T.A., M.U.A., A.Z. and S.W.L.; Funding acquisition: S.W.L.; Investigation, M.U.A. and A.Z.; Methodology, F.U.R.S., T.A. and M.U.A.; Project administration: S.W.L.; Resources, S.W.L.; Validation, S.A.; Writing—original draft: F.U.R.S.; Writing—review & editing: M.U.A., A.Z. and S.W.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by National Research Foundation (NRF) grants funded by the Ministry of Science and ICT (MSIT) and Ministry of Education (MOE), Republic of Korea (NRF[2021-R1-I1A2(059735)]; RS[2024-0040(5650)]; RS[2024-0044(0881)]; RS[2019-II19(0421)]).

Data Availability Statement: The data used were sourced from a publicly available dataset on Kaggle, under the title of "Operation of DC motor". (https://www.kaggle.com/datasets/brunaostii/funcionamento-de-um-motor-de-corrente-contnua/data, accessed on 17 July 2024).

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

References

1. Riasat, U.; Mughal, M.A.; Shoukat, A.; Ahmad, S. PMDC Motor Parameter Estimation Using Rao-1 Algorithm. In Proceedings of the 2022 International Conference on Technology and Policy in Energy and Electric Power (ICT-PEP), Jakarta International Expo, Jakarta, Indonesia, 18 October 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 145–149.
2. Nayak, B.; Sahu, S. Parameter estimation of DC motor through whale optimization algorithm. *Int. J. Power Electron. Drive Syst.* **2019**, *10*, 83. [CrossRef]
3. Alkamachi, A. Permanent magnet DC motor (PMDC) model identification and controller design. *J. Electr. Eng.* **2019**, *70*, 303–309. [CrossRef]
4. Awoda, M.L.; Ali, R.S. Parameter Estimation of a Permanent Magnetic DC Motor. *Iraqi J. Electr. Electron. Eng.* **2019**, *15*, 28–36. [CrossRef]
5. Obeidat, M.A.; Lin, F. Real-time parameter estimation of PMDC motors using quantized sensors. *IEEE Trans. Veh. Technol.* **2013**, *62*, 2977–2986. [CrossRef]
6. Obeidat, M.A.; Lin, F.; Wang, L. Online parameter estimation of PMDC motors using quantized output observations. In Proceedings of the 2012 IEEE Transportation Electrification Conference and Expo (ITEC), Dearborn, MI, USA, 18–20 June 2012; IEEE: Piscataway, NJ, USA, 2012; pp. 1–6.
7. Ghosh, M.; Ghosh, S.; Saha, P.K.; Panda, G.K. Sensorless speed estimation of permanent magnet DC brushed motor considering the effect of armature teeth–slots and commutation. *IET Power Electron.* **2017**, *10*, 1550–1555. [CrossRef]
8. Ali, A.; Mid, E.C.; Ahamad, N.; Ruslan, E.; Hadi, D. Parameter Estimation of DC Motor using Multiparametric Programming. In *Proceedings of the Journal of Physics: Conference Series*; IOP Publishing: Bristol, UK, 2022; Volume 2312, p. 012035.
9. Usman, H.M.; Mukhopadhyay, S.; Rehman, H. Permanent magnet DC motor parameters estimation via universal adaptive stabilization. *Control. Eng. Pract.* **2019**, *90*, 50–62. [CrossRef]
10. Sankardoss, V.; Geethanjali, P. Parameter estimation and speed control of a PMDC motor used in wheelchair. *Energy Procedia* **2017**, *117*, 345–352. [CrossRef]
11. Rodríguez-Abreo, O.; Rodríguez-Reséndiz, J.; Álvarez-Alvarado, J.M.; García-Cerezo, A. Meta-heuristic parameter identification of motors using dynamic response relations. *Sensors* **2022**, *22*, 4050. [CrossRef]
12. Fazdi, M.F.; Hsueh, P.-W. Parameters Identification of a Permanent Magnet DC Motor: A Review. *Electronics* **2023**, *12*, 2559. [CrossRef]
13. Czerwinski, D.; Gęca, J.; Kolano, K. Machine learning for sensorless temperature estimation of a BLDC motor. *Sensors* **2021**, *21*, 4655. [CrossRef]
14. Shanmuga, N.; Mythile, A.; Pavithra, S.; Nivetha, N. Parameter identification of a DC motor. *Int. J. Sci. Technol. Res.* **2020**, *9*, 5746–5755.
15. Mellah, H.; Hemsas, K.E.; Taleb, R. Intelligent sensor based Bayesian neural network for combined parameters and states estimation of a brushed dc motor. *arXiv* **2019**, arXiv:1901.08661. [CrossRef]
16. Sabanci, K. Artificial intelligence based power consumption estimation of two-phase brushless DC motor according to FEA parametric simulation. *Measurement* **2020**, *155*, 107553. [CrossRef]
17. Zhang, Y.; Gono, R.; Jasiński, M. An Improvement in Dynamic Behavior of Single Phase PM Brushless DC Motor Using Deep Neural Network and Mixture of Experts. *IEEE Access* **2023**, *11*, 64260–64271. [CrossRef]
18. Setiawan, M.H.; Maarif, A.; Rekik, C.; Abougarair, A.J.; Mekonnen, A.M. Enhancing Speed Estimation in DC Motors using the Kalman Filter Method: A Comprehensive Analysis. *J. Ilm. Tek. Elektro Komput. Dan Inform.* **2024**, *10*, 30–38. [CrossRef]
19. Tripathi, R.P.; Singh, A.K.; Gangwar, P. Fractional order adaptive Kalman filter for sensorless speed control of DC motor. *Int. J. Electron.* **2023**, *110*, 373–390. [CrossRef]
20. Gamazo-Real, J.-C.; Martínez-Martínez, V.; Gomez-Gil, J. ANN-based position and speed sensorless estimation for BLDC motors. *Measurement* **2022**, *188*, 110602. [CrossRef]
21. Yilmaz, A.; Simsek, C.; Balci, S. Parametric dataset-based formulation of rated torque of a brushless DC motor for electric vehicle. *Electr. Eng.* **2024**, *106*, 4327–4337. [CrossRef]
22. Unlersen, M.F.; Balci, S.; Aslan, M.F.; Sabanci, K. The speed estimation via BiLSTM based network of a BLDC motor drive for fan applications. *Arab. J. Sci. Eng.* **2022**, *47*, 2639–2648. [CrossRef]
23. İnan, R.; Aksoy, B.; Salman, O.K.M. Estimation performance of the novel hybrid estimator based on machine learning and extended Kalman filter proposed for speed-sensorless direct torque control of brushless direct current motor. *Eng. Appl. Artif. Intell.* **2023**, *126*, 107083. [CrossRef]
24. Munciño, D.M.; Damian-Ramírez, E.A.; Cruz-Fernández, M.; Montoya-Santiyanes, L.A.; Rodríguez-Reséndiz, J. Metaheuristic and Heuristic Algorithms-Based Identification Parameters of a Direct Current Motor. *Algorithms* **2024**, *17*, 209. [CrossRef]

25. Kaggle, Operation of a DC Motor. Available online: https://www.kaggle.com/datasets/brunaostii/funcionamento-de-um-motor-de-corrente-contnua/data (accessed on 17 July 2024).
26. Wu, Y.C.; Feng, J.W. Development and application of artificial neural network. *Wirel. Pers. Commun.* **2018**, *102*, 1645–1656. [CrossRef]
27. Abiodun, O.I.; Jantan, A.; Omolara, A.E.; Dada, K.V.; Mohamed, N.A.; Arshad, H. State-of-the-art in artificial neural network applications: A survey. *Heliyon* **2018**, *4*, e00938. [CrossRef] [PubMed]
28. Zou, J.; Han, Y.; So, S.S. Overview of artificial neural networks. In *Artificial Neural Networks: Methods and Applications*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 14–22.
29. Gupta, N. Artificial neural network. *Netw. Complex Syst.* **2013**, *3*, 24–28.
30. Zhang, Z.; Zhang, Z. Artificial neural network. In *Multivariate Time Series Analysis in Climate and Environmental Research*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 1–35.
31. Zhou, X.; Heng, Q.; Chi, E.C.; Zhou, H. *Proximal MCMC for Bayesian Inference of Constrained and Regularized Estimation*; The American Statistician: Alexandria, VA, USA, 2024; pp. 1–12.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.