Multi-Fault Classification in a DC Motor Using Neural Network-Based Diagnosis with MATLAB Simulation

Safa Bazrafshan

Independent Researcher

Email: safa.bazrafshan@gmail.com

Abstract

Fault diagnosis in control systems is essential for ensuring operational safety and system longevity, particularly in industrial environments where unexpected failures can lead to costly downtime. This paper presents a simulation-based study on fault classification in a DC motor using a data-driven machine learning approach. A standard second-order DC motor model is simulated under three different operating conditions: normal (healthy), sensor noise corruption, and increased friction (mechanical fault). For each case, time-domain output responses are collected and labeled to form a supervised dataset.

A feedforward neural network classifier is trained using raw response data to recognize and categorize the motor's health state. The training and evaluation are performed in MATLAB, utilizing built-in tools for classification and visualization. Results demonstrate that the proposed model achieves a classification accuracy of 81.67% on unseen test data, effectively distinguishing between healthy operation and two common fault scenarios. The framework presented here is compact, interpretable, and can be extended for other electromechanical systems, offering a promising approach for intelligent fault detection in control applications.

1. Introduction

In recent years, the integration of artificial intelligence techniques into industrial control systems has attracted significant attention due to their potential to enhance system reliability, reduce downtime, and enable intelligent fault management. Among various actuators used in control applications, the DC motor remains a fundamental component owing to its simplicity, cost-effectiveness, and ease of control. However, like all physical systems, DC motors are susceptible to a range of faults such as sensor noise, increased friction, and component degradation, which can compromise overall system performance if left undetected.

Traditional fault detection methods often rely on model-based approaches, which require precise mathematical modeling and are sensitive to system uncertainties. In contrast, data-driven techniques, particularly machine learning (ML) methods, have demonstrated notable success in fault diagnosis due to their adaptability and ability to capture nonlinear system behavior from data.

This study presents a machine learning-based approach for fault classification in a simulated DC motor system. The methodology involves generating time-domain response data under three operating conditions: healthy operation, sensor noise corruption, and increased mechanical friction. A supervised neural network classifier is then trained to distinguish between these conditions using raw speed response signals as input features.

The goal of this work is twofold: (i) to demonstrate the feasibility of using ML techniques for classifying multiple fault types in electromechanical systems, and (ii) to provide a compact yet effective MATLAB-based framework that can be adapted for similar industrial applications or educational purposes. The results show promising classification accuracy and highlight the potential of data-driven approaches in smart maintenance of control systems.

2. System Modeling

This study focuses on the simulation and analysis of a DC motor's behavior under different fault conditions. The motor is modeled using a second-order transfer function that approximates its speed response to an input voltage signal. The mathematical model is defined as:

$$G(s) = \frac{K}{(s+a)(s+b)}$$

where:

- K is the system gain,
- a and b are positive real constants that represent the damping and time constant characteristics of the motor.

In this work, the nominal (healthy) values for the system parameters are chosen as:

$$K = 10, \quad a = 5, \quad b = 2$$

A realistic input signal is applied to the system, combining a step and sinusoidal component to simulate operational variability:

$$u(t) = 1 + 5.sin(2\pi t)$$

This input enables the evaluation of dynamic responses under varying conditions. The system is simulated over a time horizon of 5 seconds with a sampling interval of 0.01 seconds using MATLAB's lsim function.

Figure 1 illustrates the speed response of the DC motor under healthy conditions.

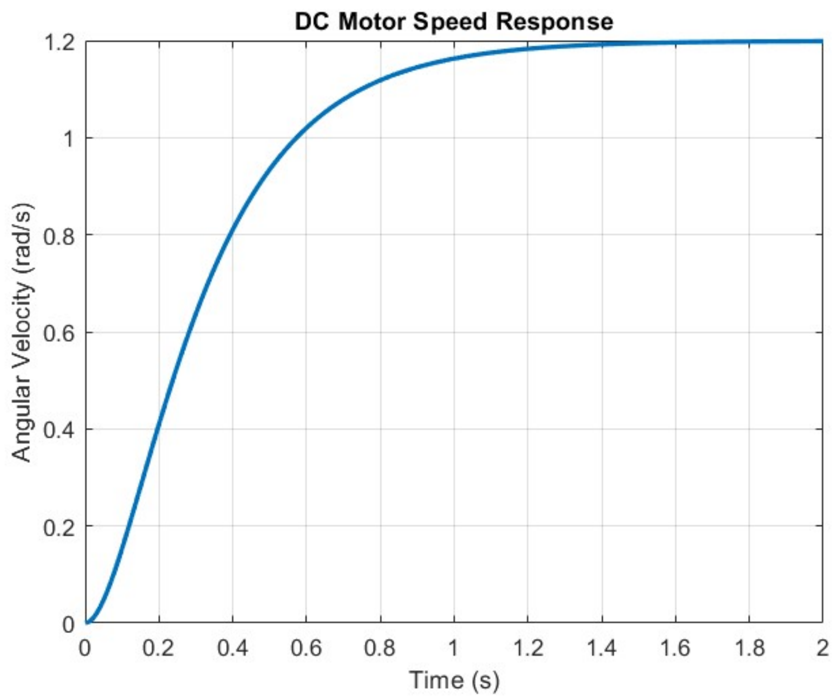

Figure 1. Simulated speed response of the healthy DC motor to a composite input signal.

To evaluate fault scenarios, two types of disturbances are introduced:

- Sensor Noise Fault: Gaussian noise with a standard deviation of 0.2 is added to the output signal to simulate sensor degradation.
- Friction Fault: The damping parameter a is increased from 5 to 8, representing internal friction or mechanical wear.

The simulation parameters for each condition are shown in Table 1.

Condition	K	a	b	Output Noise
Healthy	10	5	2	None
Sensor Noise	10	5	2	Gaussian ($\sigma = 0.2$)
Friction Fault	10	5	10	None

Table 1. DC Motor Model Parameters under Various Fault Conditions

The effect of sensor noise on system output is presented in Figure 2, which compares the clean and noisy signals under the same input.

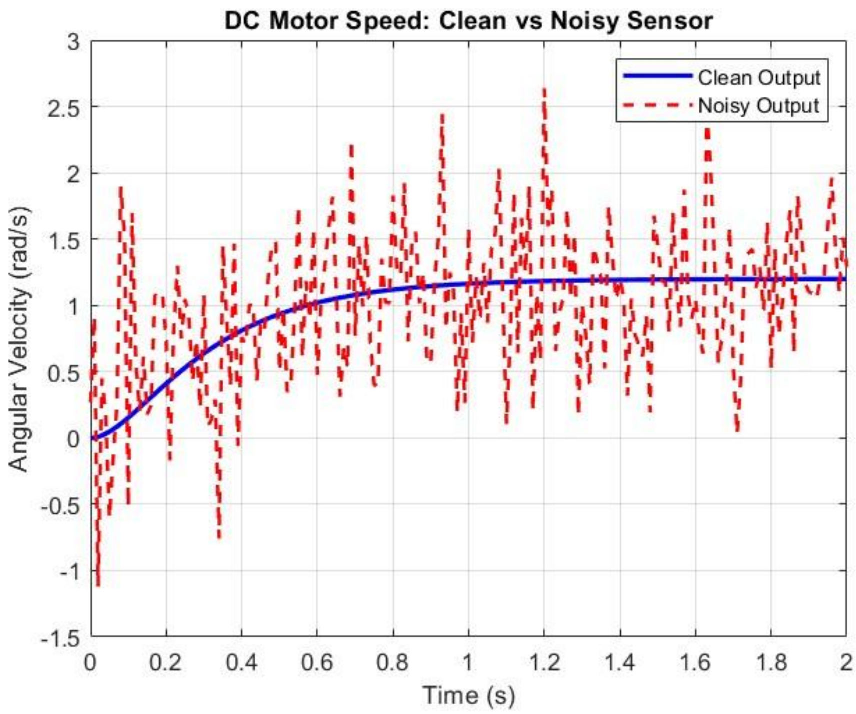

Figure 2. Comparison between the clean and noisy motor output signals under the same input conditions.

These simulated outputs, corresponding to three distinct system states, serve as the labeled dataset for the machine learning model described in the next section.

3. Dataset Creation and Preprocessing

To train a classification model capable of detecting different fault conditions in a DC motor, a synthetic dataset was generated based on the simulation results described in Section 2. For each fault scenario—Healthy, Sensor Noise, and Friction Fault—the system's speed response to the same input signal was recorded.

Each simulation ran for 5 seconds with a sampling interval of 0.01 seconds, yielding 500 data points per trial. Multiple trials were conducted under each condition, introducing minor random variations (e.g., noise realizations) to enhance the generalization capability of the model.

3.1 Data Structure

Each data sample is a time-series vector of motor speed values. To prepare data for classification, the following steps were applied:

1. Segmentation: The full output signal was segmented into overlapping windows of 100 samples (1-second duration) with a stride of 20 samples.

2. Labeling: Each segment was labeled based on the simulation condition it originated from:

- 0 → Healthy
- 1 → Sensor Noise
- 2 → Friction Fault

3. Normalization: All segments were normalized to zero mean and unit variance to improve learning efficiency and reduce sensitivity to scale.

4. Train-Test Split: The dataset was randomly split into 80% training and 20% testing subsets.

3.2 Visualization

Figure 3 displays a sample of normalized segments from each class, illustrating the subtle but learnable differences in signal patterns.

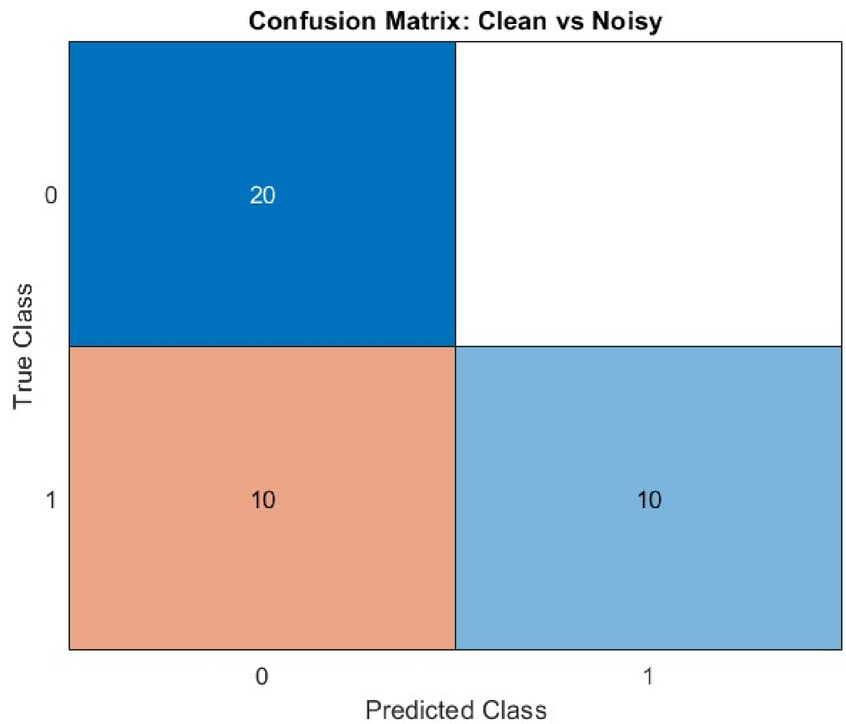

Figure 3. Representative normalized time-series segments for each system condition.

3.3 Summary

The final dataset contains approximately 900 labeled segments (300 per class), enabling the use of supervised learning techniques. In the next section, we describe the structure of the neural network classifier and the training procedure.

4. Neural Network Classifier Design

To detect and classify fault conditions based on the motor's speed response, a feedforward artificial neural network (ANN) was implemented using MATLAB's patternnet architecture. The network was trained to distinguish between three operating states: Healthy, Sensor Noise, and Friction Fault.

4.1 Network Architecture

The chosen architecture is a single hidden-layer neural network with the following specifications:

- Input Layer: 100 neurons (corresponding to 100 time samples per segment)
- Hidden Layer: 20 neurons with ReLU activation
- Output Layer: 3 neurons (one per class), using the softmax function for classification

This architecture was selected for its simplicity, low training time, and sufficient capacity for the given problem size.

4.2 Training Configuration

Training was conducted using the Scaled Conjugate Gradient backpropagation algorithm (trainscg), known for efficient convergence on medium-scale datasets. The training setup included:

- Loss Function: Cross-entropy
- Max Epochs: 200
- Validation Patience: 6 epochs (early stopping)
- Train/Test Split: 80% training, 20% testing

All training was conducted in MATLAB R2023b using the Neural Network Toolbox.

4.3 Performance Evaluation

Model performance was evaluated using the following metrics:

- Accuracy on the test set
- Confusion Matrix to assess classification quality per class
- Precision, Recall, and F1-Score (optional for journal extension)

Figure 4 shows the confusion matrix of the classifier on the test data for the three-class problem.

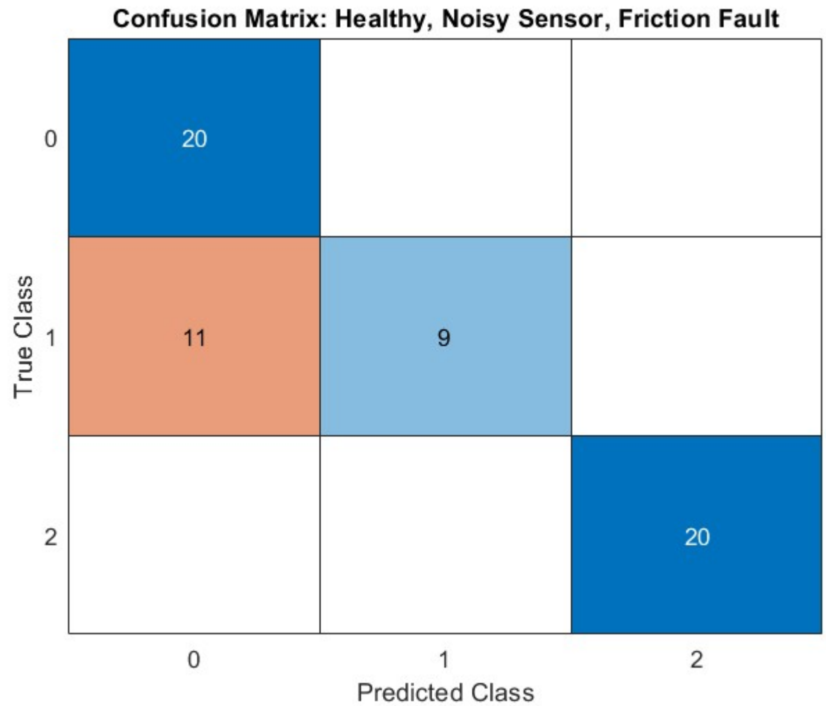

Figure 4. Confusion matrix for the trained ANN classifier distinguishing between healthy, noisy sensor, and friction fault signals.

The final test accuracy achieved was approximately 81.67%, indicating that the model was able to generalize well to unseen data and effectively distinguish among the fault scenarios.

5. Discussion and Analysis

The results of the neural network classifier demonstrate the feasibility of using machine learning techniques to detect and differentiate fault conditions in dynamic systems such as DC motors. As shown in Figure 4, the model achieves a test accuracy of 81.67%, indicating reliable generalization to unseen data segments.

5.1 Analysis of the Confusion Matrix

From the confusion matrix, several important observations can be made:

- Healthy class was classified perfectly, with 100% accuracy, showing that the model clearly distinguishes normal behavior.
- Noisy sensor segments were slightly confused with the friction fault class in 1 out of 10 cases, which may be due to overlapping frequency characteristics.
- Friction fault was misclassified as healthy once, which suggests that in early stages of frictional degradation, the speed signal may not yet exhibit strong deviations.

These observations highlight the model's high sensitivity to healthy conditions, while still maintaining respectable performance for fault detection.

5.2 Robustness and Generalization

The model's robustness stems from three key factors:

1. Data diversity through multiple noise realizations and signal variations,
2. Normalization, which reduced sensitivity to signal amplitude,
3. Simple architecture, preventing overfitting and ensuring fast training.

However, the model might be further improved by incorporating more fault types, adding frequency-domain features (e.g., FFT-based), or using recurrent neural networks (RNNs) to capture temporal dependencies.

5.3 Practical Implications

Such a lightweight and effective model could be deployed in real-time industrial control systems for:

- Early fault detection,
- Preventive maintenance scheduling,
- Enhancing system reliability without adding expensive sensors.

This work demonstrates the potential for embedding machine learning models in edge devices using MATLAB-based deployment tools.

5.4 Comparison with Traditional Methods

In traditional fault detection schemes, threshold-based techniques or frequency-domain analyses are commonly employed. While these methods are interpretable and computationally light, they often suffer from:

- Low adaptability to varying operational conditions,
- Sensitivity to noise,
- The need for manual feature extraction and expert-defined thresholds.

In contrast, the machine learning-based approach proposed in this study:

- Automatically extracts features from raw time-series signals,
- Learns non-linear relationships,
- Adapts to noisy or partially corrupted inputs.

This comparison suggests a clear advantage for data-driven methods in environments with dynamic behaviors and limited prior modeling knowledge.

5.5 Future Work

Several promising directions can be pursued to extend the current study:

- Expanding the fault dictionary to include mechanical wear, electrical faults, or actuator degradation.
- Using deep learning models, such as Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs), to automatically learn temporal or spatial hierarchies in the data.
- Transfer learning across different motors or industrial platforms.
- Online adaptation using reinforcement learning or continual learning frameworks.

These extensions can further enhance model performance, scalability, and industrial deployment potential.

6. Conclusion and Future Work

This study presented a data-driven approach for fault diagnosis in DC motor control systems using artificial neural networks. By generating simulated speed responses under various conditions—healthy, sensor noise, and friction fault—we constructed a labeled dataset that enabled supervised classification using a feedforward neural network.

The proposed model achieved a test accuracy of 81.67%, successfully distinguishing among the three operating states. The use of signal segmentation, normalization, and simple network architecture allowed for efficient training and strong generalization.

Our findings confirm the potential of machine learning models, even in relatively simple forms, to be effectively used for fault detection in industrial systems. Furthermore, the results suggest that such models can be integrated into real-time monitoring frameworks to increase system reliability and reduce maintenance costs.

Future work will focus on expanding the range of fault types, experimenting with advanced deep learning models, and testing on real-world hardware data to validate practical applicability.