electronics

Article

# Classification of Optoelectronic Rotary Encoder Faults Based on Deep Learning Methods in Permanent Magnet Synchronous Motor Drive System

**Kamila Jankowska and Mateusz Dybkowski ***

Department of Electrical Machines, Drives and Measurements, Wroclaw University of Science and Technology, 50-370 Wroclaw, Poland; kamila.jankowska@pwr.edu.pl
* Correspondence: mateusz.dybkowski@pwr.edu.pl

**Abstract:** This article presents the classification of optoelectronics encoder faults in a permanent magnet synchronous motor (PMSM) drive system. This paper proposes the deep neural networks (DNNs) speed sensor faults classification application in the vector-controlled PMSM drive. This approach to the issue has not been discussed in the literature before. This work presents a solution based on early detection with the use of the model reference adaptive system (MRAS) estimator and fault classification based on artificial intelligence. The innovative nature of this work is also due to the simulation of speed sensor damage using the developed optoelectronics encoder model in the Matlab/Simulink environment. This work is focused on simulation studies, which have been supported by experimental results obtained on the MicroLabBox platform. This article compares two structures of deep neural networks in fault detection. The results were also compared with previous experimental studies on the classification of speed sensor failures using shallow neural networks.

**Keywords:** DNN; 2D-CNN; speed sensor; FTC; PMSM



**Citation:** Jankowska, K.; Dybkowski, M. Classification of Optoelectronic Rotary Encoder Faults Based on Deep Learning Methods in Permanent Magnet Synchronous Motor Drive System. *Electronics* **2023**, *12*, 4184. https://doi.org/10.3390/electronics12194184

Academic Editors: Fulai Zhao and Fei Zhang

Received: 30 August 2023
Revised: 6 October 2023
Accepted: 7 October 2023
Published: 9 October 2023



## 1. Introduction

Currently, AC drives diagnostics [1] are met with great interest in research centers [1,2]. This is due to the use of AC motors [3] in many sensitive industries wherein failure can be a threat and lead to serious consequences [4,5]. Both the motor and drive control system are subject to diagnostics. A significant number of research works focus on detecting and classifying damages to the stator [6–8] or motor bearings [9–11]. The topic of frequency converter diagnostics is also well-developed and described in the literature [12–14]. Currently, however, an important trend is the diagnosis of damage in measurement sensors. Measuring sensors are necessary elements for the operation of the drive system in the vector control structure [15]. Additionally, sensors are widely used in the fault detection of other components of the drive system. In the case of a permanent magnet synchronous motor (PMSM), three types of sensors are used: speed/position, currents, and voltage. One of the most important listed here is the speed sensor. Damage to the remaining sensors reduces the quality of the drive system operation and, possibly, in the most important failures, may lead to a shutdown of the system [16]. This situation is different with the speed sensor. In the case of any fault, the drive system is not able to work correctly and a significant threat may appear. For this reason, many research papers describe this issue only in simulation studies.

In control systems with a PMSM, an optoelectronics encoder is most often used to determine the position of the shaft. The encoder uses digital signals, which makes it highly resistant to signal interference, but it does not allow for such high accuracy as in the case of analog measurement. However, this does not mean that such a transducer is not exposed to other types of failures, which result mainly from its construction. In research works on speed sensor damage, the sensor type is usually not specified and faults that may apply to both analog and digital signals are considered [17,18]. Article [17] shows the detection of a constant gain error, without explaining the cause of the failure and the type of sensor considered. The same type of failure is described in the article [18]. In this work, the authors developed an encoder model to simulate a failure in a way that is as close as possible to the failure of a real element, which significantly distinguishes their work from others in the literature. In addition, the value of the measured speed is much closer to the real value than in the case of simulations that do not use the speed sensor model.

The detection of speed sensor damages is mainly described by using a comparative analysis of the measured and estimated value of the speed [17–23]. Various types of estimators are used for this purpose. However, it is difficult to find works that describe the detection of speed sensor faults in a wide range of drive operating conditions and with various types of simulated damages. Paper [17] presents the detection of two types of speed sensor faults: measurement noise and offset. The detection system is based on the sliding mode observer. Only simulation results are shown here and failure detection performance is presented without detailed statistical data. Another example of this type of solution is presented in [18]. Also, in this case, the effectiveness of the fault detector is supported only by simulation results. The observer used in that work is the unknown input observer (UIO). Only gain error failure is investigated. However, it is not specified what kind of speed sensor the authors are considering in the paper. A slightly different approach is presented in [19], which describes a detection strategy based on the computation of the energy of the average standard deviation of speed data. The authors described the detection of tachometric faults (offset fault, uncertain measurement fault, zero feedback fault). Much more interesting results are reported in the work [20]. Experimental results are presented and four types of failures are considered. In addition, each of them is described. The higher-order sliding mode observer (HOSMO) was used as the speed estimator. Simulation studies of speed sensor damage detection using the model reference adaptive system observer and fuzzy logic controller are described in [22]. The use of the Luenberger observer (LO) is described in the paper [23]. The advantage of that work is that it determines the type of speed sensor (encoder) under consideration and explains the basis of the possibility of failure. The loss of information by the encoder as a result of diode damage caused by improper power supply to the encoder is considered. However, the paper presents only simulation results.

Detection using the object model requires prior knowledge of the motor parameters and does not provide damage classification. Determining the type of fault may be extremely useful, in particular when an incorrect measurement results from the improper parameterization of the measured signal. For this reason, it is necessary to use artificial intelligence methods.

In the literature, works can be found wherein the use of artificial intelligence methods in detecting damage to the stator [6,24,25], rotor [26,27], and bearings [28,29] are described. Article [6] presents stator winding fault detection and classification based on a bispectrum analysis and CNN. Another solution to this problem based on raw signals and a CNN is presented in the paper [25]. Another approach to preliminary data processing is described in [28], which addresses bearing diagnostics. Filtering, decimation, and normalization were used. 1D-CNN was used for classification. AI methods allow for achieving very high efficiency in diagnosing damage in the motor itself. However, there is little research on measuring sensors' fault detection using advanced methods. In the case of damage to the current sensor, several solutions using shallow neural networks have been presented [30–32] with the use of the multilayer perceptron (MLP). The papers [30,31] present only simulation results, while in the paper [32] the results are also supported by experiments. The detection of speed sensor damage, also using the MLP, is described in the works [33,34]. These works present simulation and experimental results. These results are satisfactory; however, it is worth working on the use of currently more popular deep learning methods to improve the efficiency of speed sensor fault classification.

The basis of this article is the development of the encoder model in the Matlab/Simulink environment. This made it possible to simulate fault in a manner not previously presented in the literature. This approach made it possible to simulate the loss of pulse failure in individual encoder channels A and B. In general, three types of faults are considered in this paper: loss of pulses, software scaling error of the signal, and total loss of the signal. The fault monitoring system is based on a detector and classifier. The MRAS estimator was used for initial damage detection. After detecting a fault, the system is switched to sensorless mode. Deep neural networks (DNNs) were used as damage classifiers. The results are presented for two types of networks, namely a simpler version without the use of convolutional layers and a more complex classifier with five convolutional layers, wherein the innovative nature of this work also lies. There are no works in the literature that present the classification of damage in measurement sensors using DNNs. This work presents detailed statistical analyses of the obtained results and input vectors. Usually, statistical analysis is not presented in works on similar topics. As in the previous work, the authors present the use of shallow neural networks in the classification of speed sensor failures [33]; this work also shows a percentage comparison of both methods.

This work is organized as follows. The first chapter describes the review of the literature and the innovativeness of this approach. The second part presents the developed model of the optoelectronics encoder in simulation studies. The failure simulation method using the encoder model directly is also presented in the second section. The third chapter is a description of the developed fault classifier based on DNNs. Selected structures, input and output vectors, and training methods are described. The following chapters contain the classification of faults based on DNNs, successively in simulation and in experimental tests. Statistical data and examples of the application's operation during offline classification are presented. The last part is a summary of the obtained results.

## 2. Optoelectronic Encoder Model

### *2.1. Modeling of the Optoelectronic Encoder in Simulink*

This chapter presents the optoelectronics encoder model used in simulation studies. The model was developed on the basis of the works [35,36]. The main components of the incremental encoder are the rotating code disc and the optical system, containing transmitters–LEDs and receivers–phototransistors (Figure 1) [37,38]. The increments applied to the code disc are laser-cut spaces in the metal disc through which the light of the transmitter passes. The number of marked increments defines the encoder resolution parameter. The model presented in this research is based on a resolution of 3600 impulses per rotation. This value was chosen because it is a commonly used value.

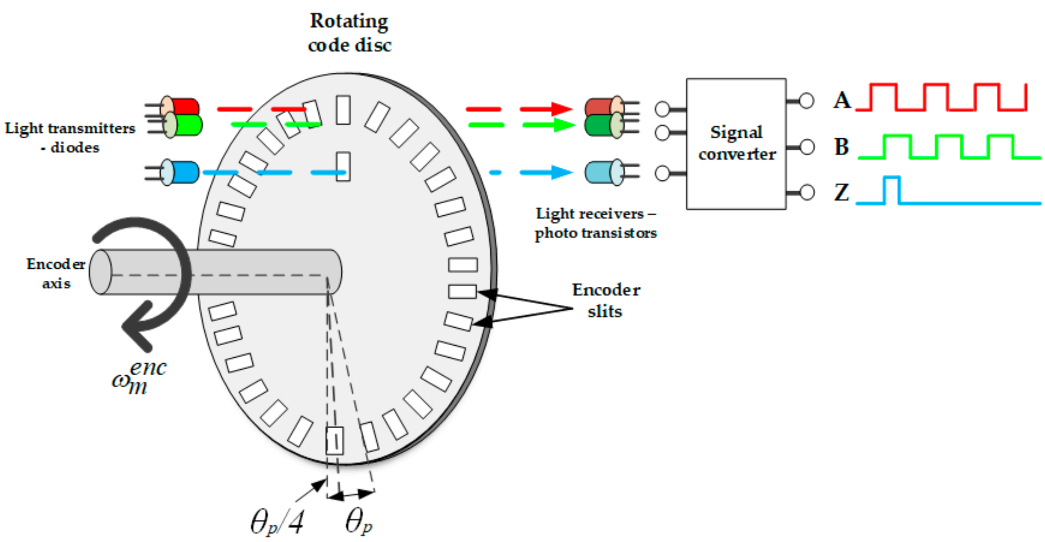


**Figure 1.** Incremental encoder schema.

The position of the motor shaft is calculated based on the two basic encoder channels A and B, which are shifted in phase by 90 degrees. The simultaneous use of channels A and B enables the identification of the rotation direction of the shaft. The controller detects which signal is reported first. For this purpose, appropriate logical determinants are used. If channel A is reported first, then the encoder rotates clockwise; if channel B is the first, it means that the direction of rotation has been changed to counterclockwise. All possible logical notations used to determine the direction of the shaft are presented in Table 1. An additional signal (Z) is generated after each full rotation. The third channel is called a synchronization signal. The signals logic from the optoelectronic system is defined as follows:

$$A(\theta) = \begin{cases} 1, & 0 < \theta \mathrm{mod}(\theta_p) \leq \frac{\theta_p}{2} \\ 0, & \frac{\theta_p}{2} < \theta \mathrm{mod}(\theta_p) \leq \theta_p \end{cases}, \tag{1}$$

$$B(\theta) = \begin{cases} 1, & 0 < (\theta - \frac{\theta_p}{4})\mathrm{mod}(\theta_p) \leq \frac{\theta_p}{2} \\ 0, & \frac{\theta_p}{2} < (\theta - \frac{\theta_p}{4})\mathrm{mod}(\theta_p) \leq \theta_p \end{cases}, \tag{2}$$

$$Z(\theta) = \begin{cases} 1, & \theta \mathrm{mod}(2\pi) = 0 \\ 0, & \theta \mathrm{mod}(2\pi) \neq 0 \end{cases}, \tag{3}$$

where θ—angular position, $\theta_p$—angular step of encoder.

**Table 1.** Counterclockwise (CCW) and clockwise (CW) logic, where 0→1—raising edge, 1→0—falling edge.

| From CCW to CW Logic | | From CW to CCW Logic | |
|---|---|---|---|
| A | B | A | B |
| 0 | 0→1 | 0→1 | 0 |
| 0→1 | 1 | 0 | 1→0 |
| 1→0 | 0 | 1 | 0→1 |
| 1 | 1→0 | 1→0 | 1 |

The developed encoder model can be divided into four main parts (Figure 2):

– A, B, Z signal generation block (1–3);
– Position determination block (4);
– Speed determination block (5);
– Direction identification block based on logic presented in Table 1.

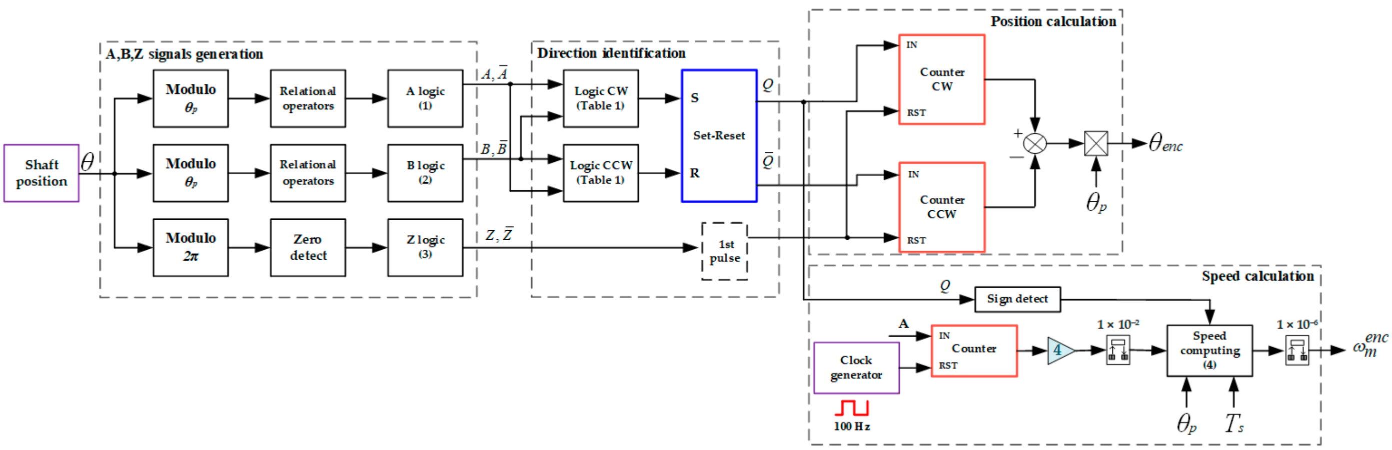


**Figure 2.** Incremental encoder general block diagram.

The A, B, Z signal generation block is based on the shaft position determined from the PMSM model in Simulink and Equations (1)–(3). Next, the direction of rotation is determined. Here, the logic shown in Table 1 and the Set-Reset block were used.

Logical notations presented in Table 1 enable the generation of pulses defining a specific direction of rotation, which in the next step are counted with the use of a Simulink counter and summed up accordingly. The impulse from the Z channel after a full revolution is responsible for resetting the counter. This makes it possible to determine the shaft position based on the following equation:

$$\theta_{enc} = \theta_p\left(\sum_{i}^{CCW} N_i - \sum_{j}^{CW} N_j\right) = \theta_p N \tag{4}$$

where

$\theta_p = \frac{2\pi}{3600}$—encoder graduation based on encoder resolution 3600;
$N$—number of counted impulses.

Calculation of the speed with the use of an encoder consists in counting the appearing impulses in any channel in a specific time interval—$T_s$. In this research, it was decided to calculate the number of pulses on channel A. Determining the speed based on the generated pulses must be conducted with a higher sampling period ($1 \times 10^{-2}$) than the entirety of the simulations ($1 \times 10^{-6}$). Otherwise, pulse counting would not be possible. At most, one pulse could occur in one sample. In the speed determination block, encoder speed is calculated based on the following equation:

$$\omega_m^{enc} = \frac{\theta_p \Delta N}{T_s} \tag{5}$$

where

$\theta_p = \frac{2\pi}{3600}$—encoder graduation based on encoder resolution 3600;
$\Delta N$—number of counted impulses in a defined period;
$T_s = 10^{-2}$—encoder sampling period.

Example waveforms of signals from channels A, B, and Z are shown to confirm their correct generation on the basis of the shaft position in Simulink (Figure 3). Signals A and B are 90 degrees out of phase. The waveforms obtained from the direction determination block are also shown below (1—clockwise, 0—counterclockwise).

The waveforms comparing the speed and position of the motor determined from the PMSM model in Simulink and using the encoder model are shown in Figure 4. The results are presented for both directions of rotation. The speed curves are shown after applying a filter $\frac{1}{1+0.005s}$ to minimize the quantization effect. The lowest velocity determination accuracy appears for the lowest speeds. In the case of position calculation, the quantization effect is hardly noticeable at the selected resolution of 3600.

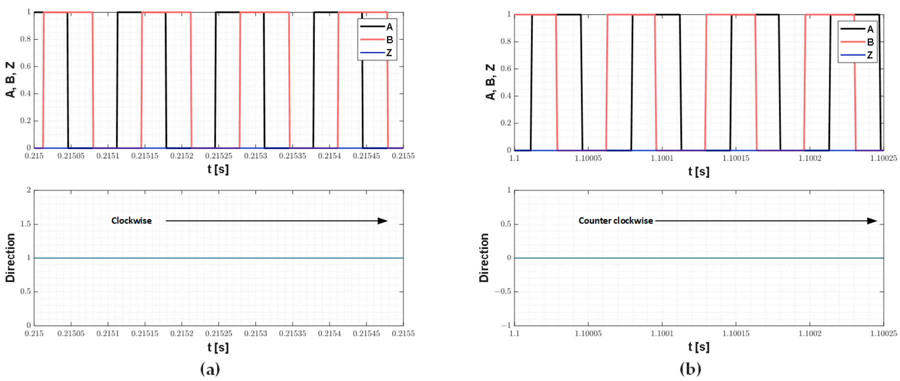


**Figure 3.** *Cont.*

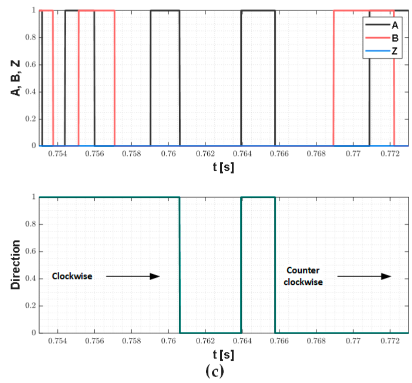


**Figure 3.** A, B, Z signal transients and direction identification block output during positive (**a**), negative speed value (**b**), and changing direction (**c**).

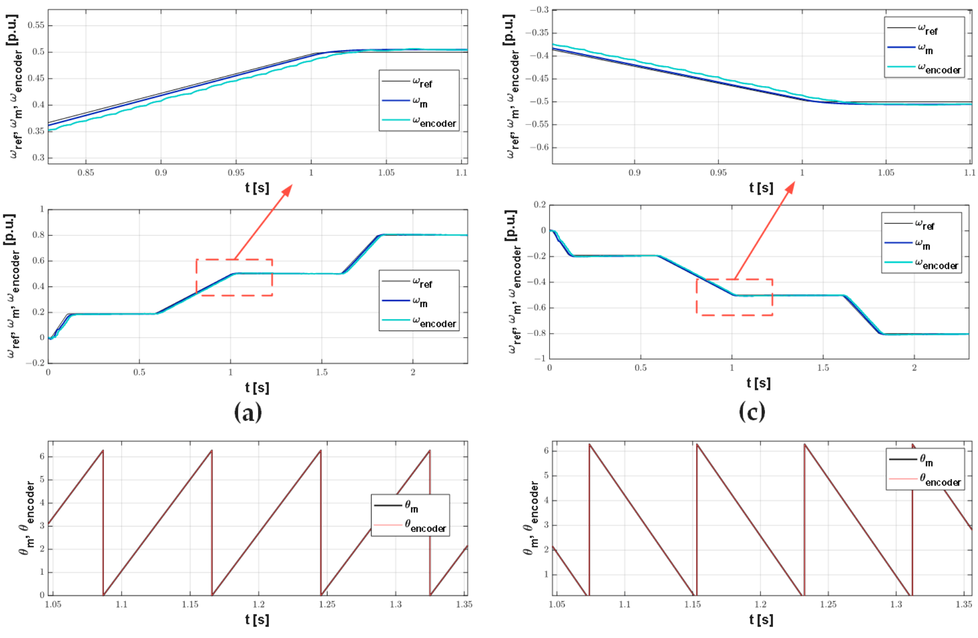


**Figure 4.** Speed and angular position transients based on developed encoder model for positive (**a**,**b**) and negative speed value (**c**,**d**).

### *2.2. Fault Simulation with the Use of a Developed Encoder Model*

The encoder uses digital signals and is resistant to measurement noise. The worst damage that can appear in this type of transducer is the complete loss of the measurement signal, which can lead to a catastrophe. Such a loss may result from a damaged cable or electronic components. Another factor responsible for incorrect speed measurement may be a scaling error in the control structure. The last considered failure in operation is the loss of pulses. The development of the encoder model made it possible to simulate the loss of individual pulses in channels A and B—resulting from the blocked holes of the encoder disc. Depending on the kind of fault, it may be a signal loss only in channel A or B, or in both. A loss in one channel leads to a situation wherein only pulses in one direction are counted, which can lead to a speed calculation with the opposite sign. Such a case is shown in Figure 5. In the case of loss of pulses from both channels, the damage remains in a course of measurement noise. The impact of individual-considered faults on the speed and position transients is shown in Figure 5.

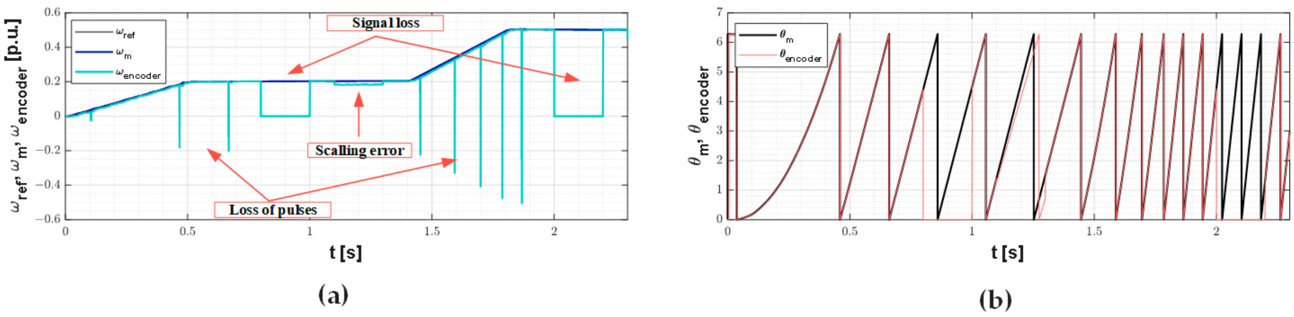


**Figure 5.** Speed (**a**) and encoder position (**b**) transients during all considered types of faults.

## 3. Speed Sensor Fault Detection and Classification Mechanism

### *3.1. Fault Detection Mechanism Based on Model Reference Adaptive System*

In the presented solution, a comparative detector based on the model reference adaptive system (MRAS) estimator is responsible for early failure detection and switching the system to sensorless mode. The detailed analysis of this type of detector was described in the authors' earlier paper [33]. This work is mainly focused on damage classification. The MRAS estimator structure is divided into three basic elements: reference model (PMSM), adjustable model, and adaptive law. The operation of the MRAS adjustable model is based on the following equation:

$$\frac{d}{dt}\begin{bmatrix}\hat{i}_d\\ \hat{i}_q\end{bmatrix} = \begin{bmatrix}\frac{-R_s}{L} & \hat{\omega}_e\\ \hat{\omega}_e & \frac{-R_s}{L}\end{bmatrix}\begin{bmatrix}\hat{i}_d\\ \hat{i}_q\end{bmatrix} + \frac{1}{L}\begin{bmatrix}U_d + \frac{R_S\psi_f}{L}\\ U_q\end{bmatrix} \tag{6}$$

where

$i_d, i_q, \hat{i}_d, \hat{i}_q$—measured and estimated currents in *d*-*q* coordinates;
$\psi_f$—magnetic flux;
$L$—stator inductance;
$R_s$—stator resistance;
$U_d$, $U_q$—voltages in *d*-*q* coordinates;
$\omega_e$, $\hat{\omega}_e$—measured and estimated electrical speed.

In MRAS, speed is determined using the adaptive law. In this research, the adaptation law defined by the following equation was selected:

$$\hat{\omega} = (K_p + \frac{K_i}{s})(i_d\hat{i}_q - i_q\hat{i}_d - \frac{\psi_f}{L}(i_q - \hat{i}_q)) \tag{7}$$

A general diagram of the MRAS speed estimator is presented in Figure 6.

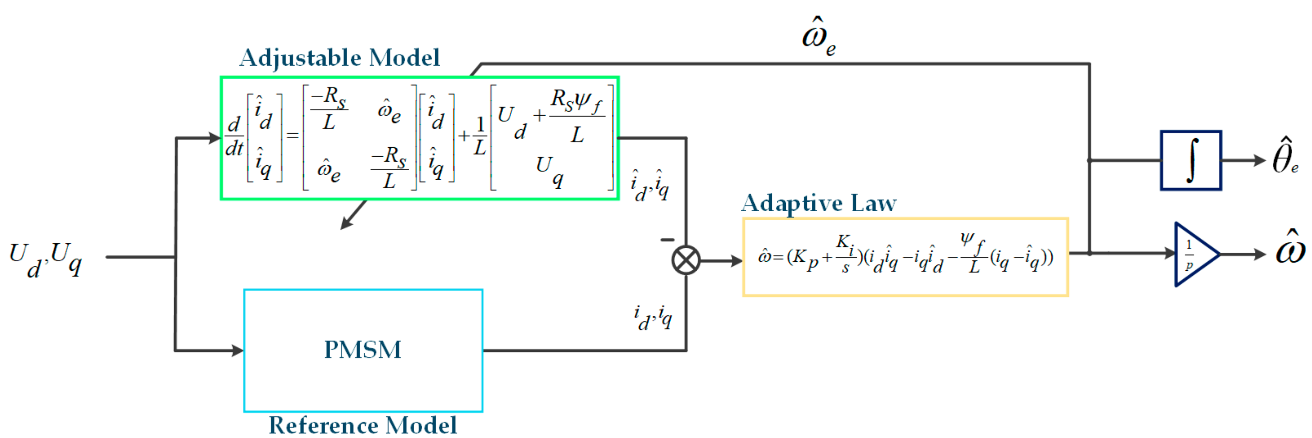


**Figure 6.** MRAS scheme including reference model, adjustable model, and adaptive law equations, where $p$ = 4.

Early damage detection is based on comparing the measured and estimated values of speed and current in the q-axis. In the fault detection system, the threshold for switching to sensorless mode is set to 0.013 for a speed error, and for a current error it is set to 0.005. The choice of these values is a compromise between fast detection and limiting false alarms about damage.

### 3.2. DNN Structures Used in Research

Machine condition monitoring based on deep neural networks is characterized by great possibilities in terms of selecting the network structure, the number and type of layers, and the parameters of the learning process. For this reason, in this study, it was decided to analyze two structures—one with convolutional layers and another simplified without these layers. Due to the complex structure of the DNN, they enable the detection of object features that would not be possible with the use of shallow neural networks, like multilayer perceptron.

In the DNN structure, we can divide the applied layers into feature detectors and classifiers. In this work, only the layers used in the research will be characterized [39]. There are no specific rules in the literature regarding the selection of neural structures. In this article, empirical tests were the basis for selecting the types and parameters of network layers. Apart from that, it was based on structures described in articles on monitoring other types of damage. In the case of the first type of classifier, the number of fully connected layers and their size had the greatest impact on the effectiveness. However, for the type II classifier, the number of convolution layers, the size of the filters, and their number had a significant impact on the effectiveness. With a smaller number of convolutional layers, the classification efficiency decreased, while increasing their number led to over-fitting to the training data and reduced the generalization properties of the neural network.

In the type I classifier, convolutional layers are not included. The overall structure is similar to the multilayer perceptron but it is more complex. The network structure consists of the following layers:

- Fully connected layers—layers responsible for multiplying the input signals by appropriate weights and adding biases—operation similar to multilayer perceptron's;
- Batch normalization layers—responsible for the standardization of the input vector to each mini-batch. The main purpose of using a layer is to stabilize the training process;
- ReLU layers—non-linear activation function defined as

$$\mathrm{f(x)} = \begin{cases} \mathrm{x}, & \mathrm{x} > 0 \\ 0, & \mathrm{x} < 0 \end{cases}; \tag{8}$$

- Softmax layer—is responsible for transforming the output vector so that the sum of the individual elements is 1. Softmax converts output values to a probability by first taking it to an exponential and then dividing by the summed exponential all the elements in the original output vector. The operation on a particular element is as follows:

$$\frac{e^{z_i}}{\sum\limits_{j=1}^{K} e^{z_j}} \tag{9}$$

where

$z_i$—individual vector element;
$K$—number of output vector elements.

- Classification layer—the cross-entropy loss for classification is computed in this layer for elements determined in the classification layer, which occurs before.

In the case of the type II classifier, convolutional and maxpooling layers were additionally used. Five convolutional layers were used with 5 × 5 filters in the amount of 30 (first layer), 60 (second and third layers), and 90 (fourth and fifth layers) for the respective layers. The structure of the type II classifier contains the following layers:

- Convolutional layers—layers containing filters whose parameters are selected during the learning process. Before the training process, the size of the filters is defined;
- Batch normalization layers;
- ReLu layers;
- Maxpooling layers—operation consisting of selecting the largest element from the feature map, covered by filters;
- Fully connected layer;
- Softmax layer;
- Classification layer.

### *3.3. Inputs and Outputs of DNN Classifiers*

The signals used as DNN inputs are the essence of proper fault classification. The input vector is based on the measured and estimated values of speed and current in the $q$-axis. The next input signals are the differences between the measured and estimated quantities using MRAS. Currents in the q-axis and their previous values are also used. The current values in the q-axis were chosen because previous research showed the high effectiveness of simple failure detectors, based on state variable observers, using this signal [33]. The full input vector for experimental (10) and simulation studies (11) is shown below:

$$\begin{array}{ccccc} [\omega_{ref}(k), & \omega_{meas}(k), & \omega_{err}(k), & \omega_{err}(k-1), & \omega_{err}(k-3), \\ \omega_{err}(k-5), & \omega_{err}(k-7), & i_{sq_{meas}}(k), & i_{sq_{err}}(k), & i_{sq_{err}}(k-1), \\ i_{sq_{err}}(k-3), & i_{sq_{err}}(k-5) & i_{sq_{err}}(k-7)] & & \end{array} \tag{10}$$

$$\begin{array}{ccccc} [\omega_{ref}(k), & \omega_{enc}(k), & \omega_{err}(k), & \omega_{err}(k-300), & \omega_{err}(k-1000), \\ \omega_{err}(k-1500), & \omega_{err}(k-2000), & i_{sq_{enc}}(k), & i_{sq_{err}}(k), & i_{sq_{err}}(k-300), \\ i_{sq_{err}}(k-1000), & i_{sq_{err}}(k-1500) & i_{sq_{err}}(k-2000)] & & \end{array} \tag{11}$$

In the case of the type I classifier, the network input is given without any transformations. The type II classifier requires transformation using the *reshape* Matlab function of individual inputs to the form [13 1 1]. This is due to the fact that the convolutional process should, by definition, be performed on matrices, not vectors.

The input vector used in simulation studies and experiments differs slightly in terms of the number of previous samples used. This is due to the fact that a different sampling time ($1 \times 10^{-6}$) was used in the simulation studies than in the experimental studies ($1 \times 10^{-4}$). Table 2 shows all classifier inputs with appropriate descriptions.

**Table 2.** Description of individual DNN classifier inputs.

| Input | Experimental Studies | Simulation Studies | Description |
|---|---|---|---|
| $\omega_{ref}(k)$ | $\omega_{ref}(k)$ | $\omega_{ref}(k)$ | Reference speed value. |
| $\omega_{meas}(k)$ | $\omega_{meas}(k)$ | $\omega_{meas}(k)$ | Measured speed value in actual sample. |
| $\omega_{err}(k-i) = \lvert\omega_{meas}(k-i) - \omega_{est_{MRAS}}(k-i)\rvert$ | $i = 1,3,5,7$ | $i = 300, 1000, 1500, 2000$ | The error between measured and estimated speed values in previous samples. |
| $i_{sq_{meas}}(k)$ | $i_{sq_{meas}}(k)$ | $i_{sq_{meas}}(k)$ | Measured and estimated q-axis current value in actual sample. |
| $i_{sq_{err}}(k-i) = \left\lvert i_{sq_{meas}}(k-i) - i_{sq_{MRAS}}(k-i)\right\rvert$ | $i = 1,3,5,7$ | $i = 300, 1000, 1500, 2000$ | The error between measured and estimated q-axis current value in previous samples. |

where $\omega_{meas}$—measured speed, $\omega_{est_{MRAS}}$—MRAS estimated speed value, $i_{sq_{meas}}$—measured $q$-axis current value, $i_{sq_{MRAS}}$—MRAS estimated $q$-axis current value.

The output of both types of classifiers is information about the type of damage to the speed sensor. The classifier output is identical for both simulation and experimental research:

0—no fault;

1—loss of pulses;

2—scaling error;

3—loss of measurement signal.

The complete structures with the input and output of both classifiers are shown in Figure 7.

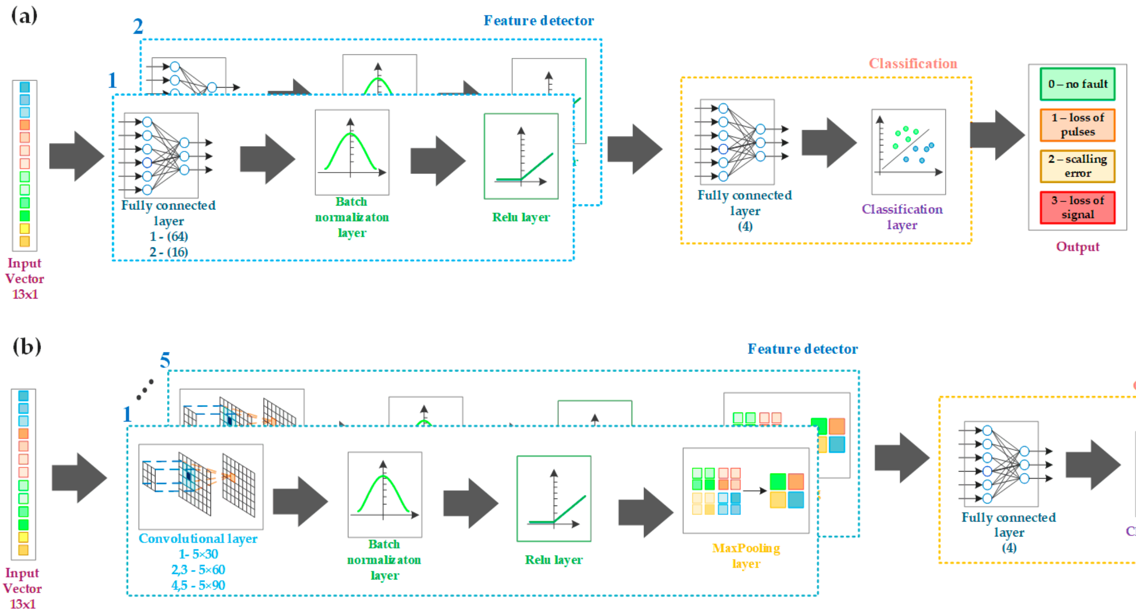


**Figure 7.** DNN structures: type I classifier (**a**), type II classifier (**b**).

### 3.4. Training Process of DNN Classifiers

SGDM (stochastic gradient descent with momentum) was chosen as the training method for both types of classifiers. When choosing this learning method, it is necessary to specify the *Initial Learning Rate*, which was selected as 0.01. The choice of *Initial Learning Rate* value was made empirically. Decreasing or increasing the value worsened the fit of the network to the training data. The *mini-batch size* was left as the default value of 128. The momentum default value of 0.9 was also left. *Training Accuracy* and *Training Loss* waveforms in the initial part of learning for simulation and experimental studies are shown in Figures 8 and 9. In both cases, a faster fit to the network training data is obtained for the type II classifier. A fit close to 100% is already obtained at about the 1500th epoch. The full learning process consisted of 62,500 iterations and lasted nearly 30 min. The time of the training process is mainly due to the number of samples in the training vector.

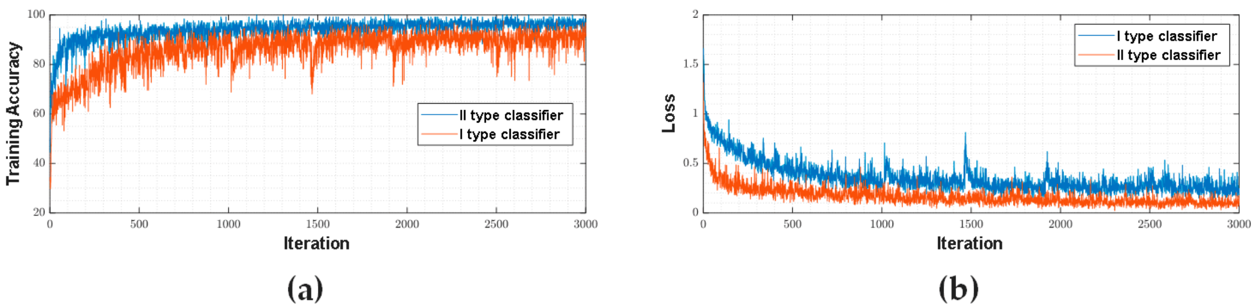


**Figure 8.** Transients of Training Accuracy (**a**) and Training Loss (**b**) for the initial part of training in simulation studies.

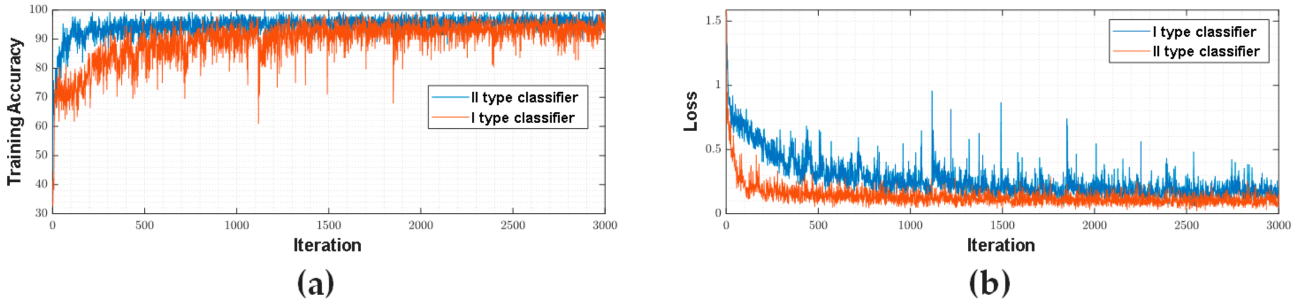


**Figure 9.** Transients of Training Accuracy (**a**) and Training Loss (**b**) for the initial part of training in experimental studies.

## 4. Simulation Results

This section describes the obtained statistical results and exemplary waveforms from simulation tests for offline damage classification. Simulation tests were carried out in the Matlab/Simulink (R2021b) environment. In this research, the Euler method with a fixed step size equal to $1 \times 10^{-6}$ s was used. In order to calculate the speed using the encoder model, $1 \times 10^{-2}$ s was set. The model of the control structure was prepared in the Sim Power System Toolbox and the neural network was designed using the Deep Learning Toolbox. The preparation of the training and testing vector was carried out in the Simulink environment, while the network training and testing process was developed in Matlab. The motor parameters used in the simulation tests are identical to the motor parameters from the experimental tests (Table 3).

**Table 3.** Parameters of the tested motor.

| $P_N$ (kW) | $P_p$ (-) | $n_N$ (rpm) | $T_N$ (Nm) | $I_N$ (A) | J (kg·m$^2$) | $R_S$ (Ω) |
|---|---|---|---|---|---|---|
| 0.894 | 4 | 6200 | 1.4 | 1.9 | 0.000039 | 4.6615 |

The input vector for the training process was based on three speed values in both directions, while the tests were carried out using courses with two constant speeds. The exact parameters of the training and testing vectors are presented in Table 4. To determine the statistical significance of the input data, the average *p*-value was determined for all observations, which is 0.0378 in the case of the training vector and 0.0358 in the case of the test vector. Assuming a significance threshold of 0.05, the obtained data are significant and the null hypothesis is rejected. The next statistical test was the determination of the correlation between individual elements of the input vector and the output vector. The results are presented in Figure 10. Based on these results, it can be concluded that the most important values in the classification of damages are the values from previous samples of both speed and current errors. Moreover, each input shows a significant association with the output vector.

**Table 4.** Training and testing data parameters in simulation studies.

| Feature | Training Data | Testing Data |
|---|---|---|
| Number of samples | 16,000,002 | 10,400,002 |
| Speed values | $\pm 0.2\omega_{ref}$, $\pm 0.5\omega_{ref}$, $\pm 0.9\omega_{ref}$ | $\pm 0.3\omega_{ref}$, $\pm 0.7\omega_{ref}$ |

Confusion matrices were used to illustrate the effectiveness of individual classifiers (Figure 11). The results show that the more complex type II classifier allows for slightly higher efficiency. This is particularly visible in the classification of the scaling error and the loss of pulses from the encoder. A significantly higher level of fit to the training data can be seen in the case of the type II classifier. This was previously also depicted by the

Training Accuracy transient (Figure 8). However, for the data selected for testing, there is no negative effect of overfitting when the testing data show a wide range of speed in both directions. Scaling error and loss of pulses from the encoder are the most complex faults to classify. Particularly in the case of loss of pulses from the encoder, the effect on the control structure is not always visible. This results in classification errors. Errors also appear in distinguishing these failures from each other. Here, samples from previous values are particularly important in the classification. Extending the training vector with additional values from previous samples leads to an increase in the efficiency of classification, at the cost of increasing the complexity of the classifier and extending the learning process.

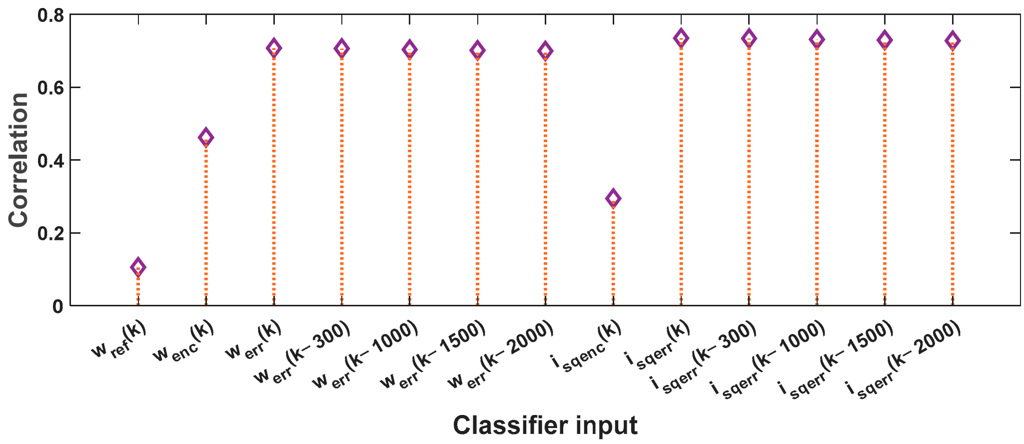


**Figure 10.** Correlation values between output vector and individual input vector elements of training data in simulation studies.


**Figure 11.** Confusion matrices for training and testing data for type I classifier (**a**,**b**) and type II classifier (**c**,**d**) in simulation studies.

Another element presenting statistical data is a comparison of the percentage results of the classifier effectiveness depending on the motor speed for the testing vector. Data are obtained for static operations only during failures, ignoring non-failure states (Figure 12). The basic conclusion that follows from the figure presented is the increase in the effectiveness of the classifier with the increase in speed. Additionally, higher efficiency is obtained for negative speed values for both types of classifiers.

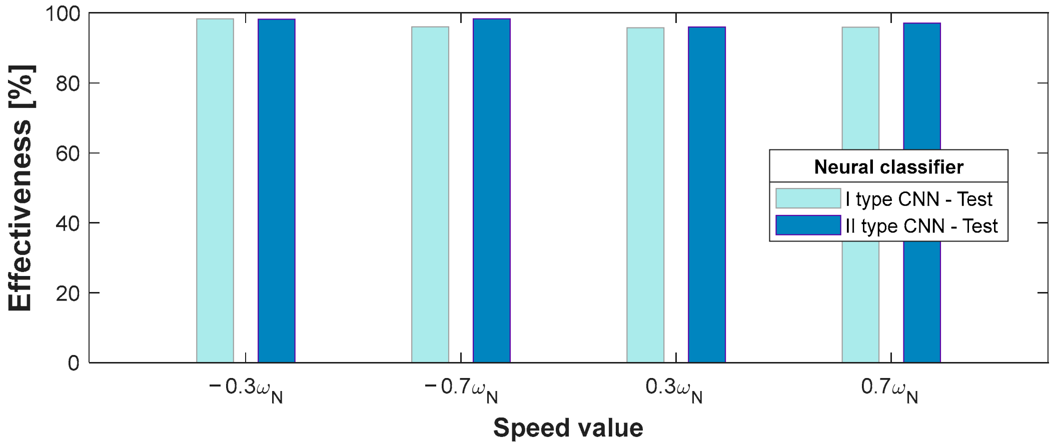


**Figure 12.** Percentage results of classifier effectiveness depending on speed in simulation studies for both types of classifier.

In the further part of this section, examples of classification of all types of damage are presented. The presented samples make it possible to draw certain conclusions as to the operation of the classifier, but they are not as meaningful as the statistical data presented using the confusion matrices. Moreover, these are waveforms without modification of the output of the neural network, e.g., by rounding the values from several samples, which will only be performed in future online classification research. First, the operation of the classifier during the loss of the encoder pulses was analyzed (Figure 13). The waveform of the detector trigger signal based on the difference between the measured and estimated speed was also presented. This is the signal responsible for switching the system permanently into sensorless mode. A loss of signal in channel A or B affects more samples than the duration of the loss itself, which both of the classifiers capture accordingly. It is also visible here that errors in distinguishing scaling failure from loss of pulses occur mainly at low speeds. In addition, it can be observed that, in the case of the type I classifier, single pulse losses are more often not detected.

Further on, the transients of the two other types of failures are presented: signal loss and scaling error (Figure 14). The simplest type of damage to classify, i.e., signal loss, is detected almost flawlessly. In waveforms with this fault, there are false fault alarms (single samples), resulting from the dynamic states of the PMSM. In the case of scaling error, there is a clear advantage in the operation of the type II classifier. In the case of the type II classifier, there are few errors during the whole presented transient. Switching on a fault in a dynamic state does not significantly affect correct classification.

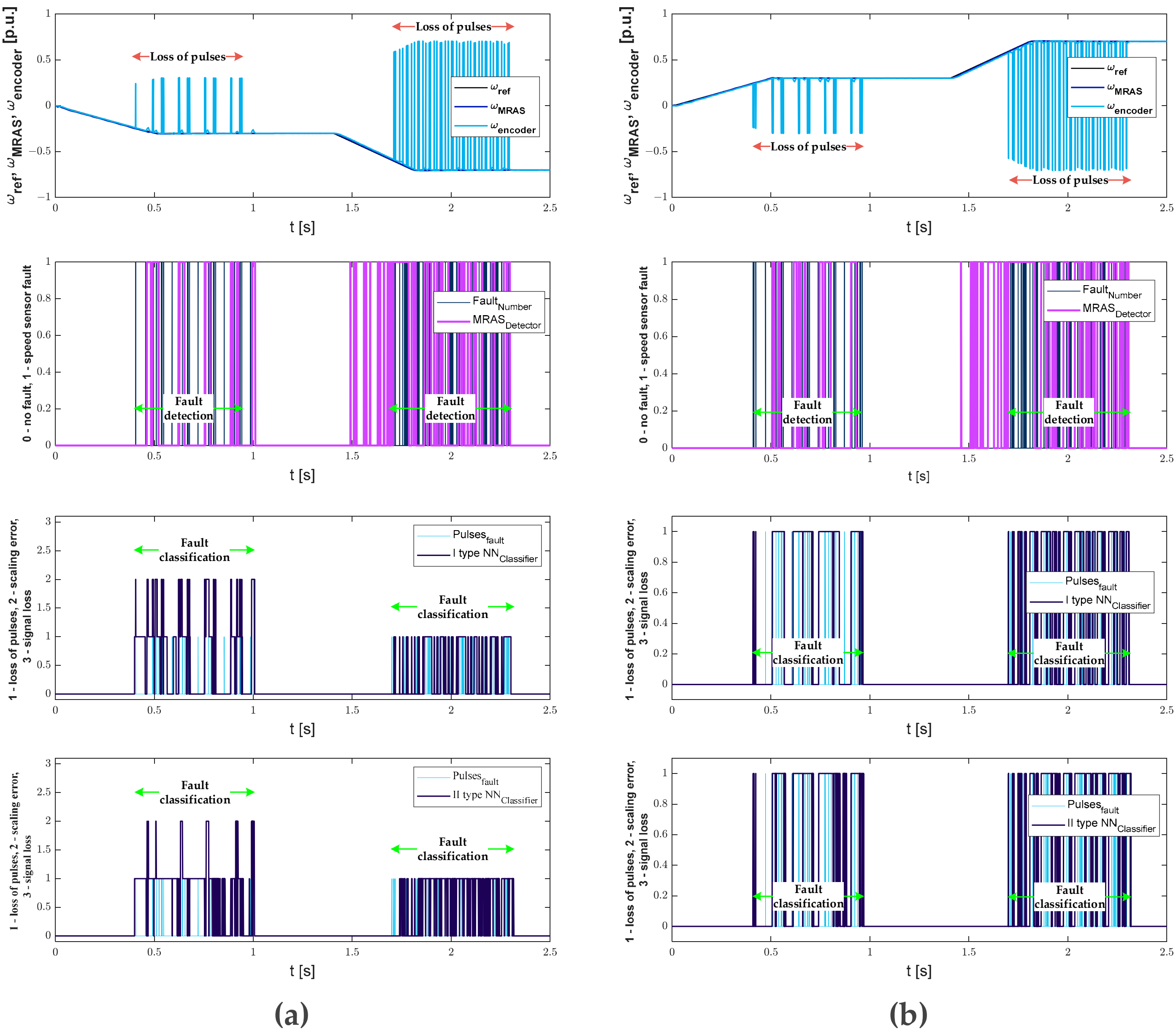


**Figure 13.** Transients speed, MRAS detector, and fault classifiers during the loss of pulses fault for negative (**a**) and positive (**b**) speed direction.

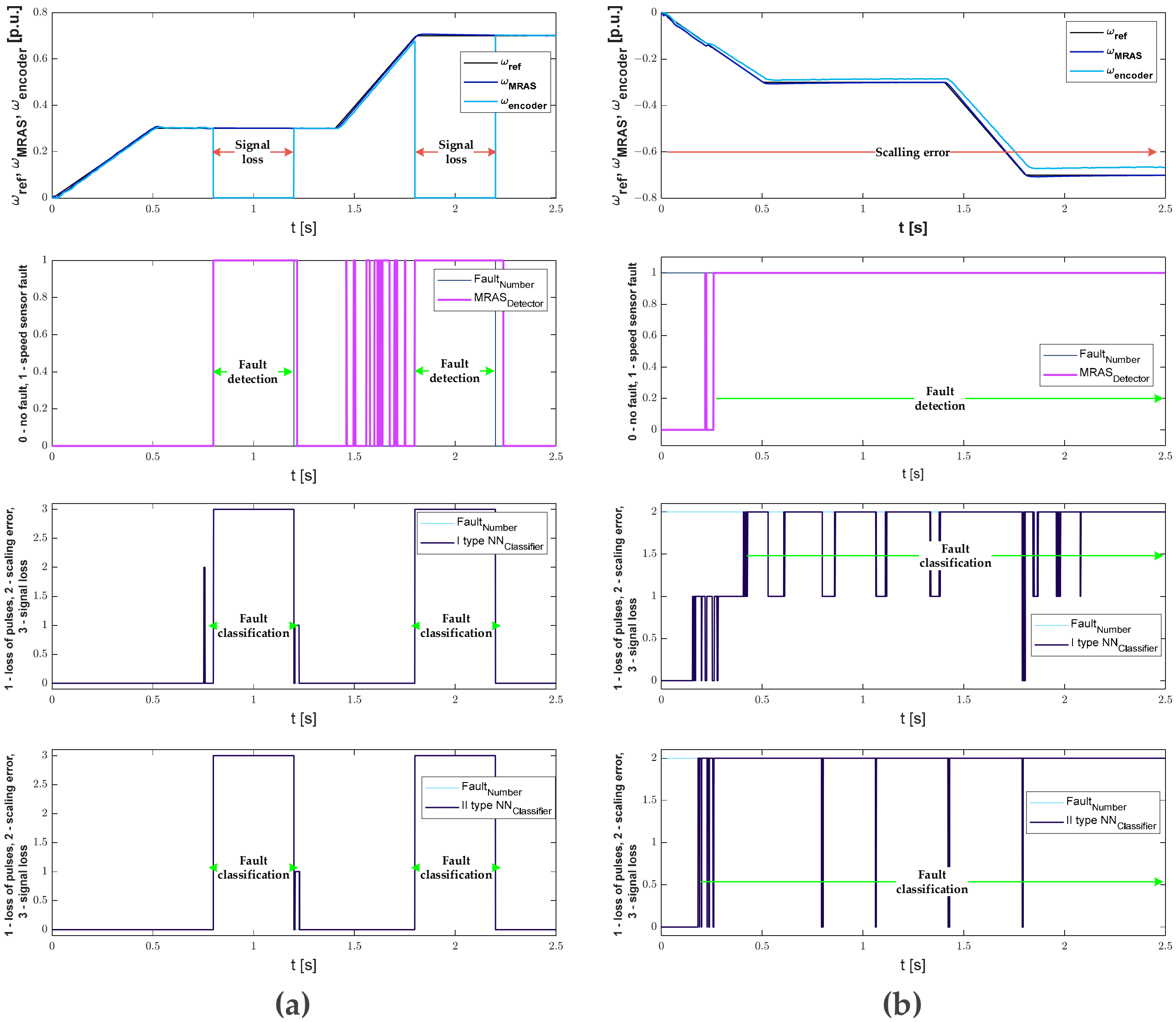


**Figure 14.** Transients speed, MRAS detector, and fault classifiers during the loss of signal fault (**a**) and scaling error (**b**).

## 5. Experimental Results

Experimental tests were carried out on a 0.894 kW PMSM motor from Moog (G403-2007A). Other motor parameters are presented in Table 3 in the simulation studies chapter. The dSpace DS1103 rapid prototyping system with Control Desk (version 6.4) and Matlab/Simulink (R2021b) software were used in the tests. Real-Time Simulink Toolbox was used to conduct experimental research. The position of the shaft was measured with an incremental encoder (36,000 imp./rev), and the current measurement was carried out using LEM-type current transducers. Another Moog PMSM motor (G404-2009A—0.89 kW, with nominal load 1.09 Nm and nominal speed value 7800 rpm) controlled by a Multi-Axis Servo Drive System was used as the load. Moog Drive Administrator 5 (5.4.52) software was used to control the test motor load. The load of the appropriate value was switched on using the torque control method. A photo of the test stand is shown in Figure 15. The parameters of the data used in the training and testing of the classifiers are presented in Table 5. Experimental studies were carried out for the same type of damage, but the loss of the pulses was simulated by adding measurement noise to the speed course, which is consistent with the A and B channels' signal loss simultaneously.

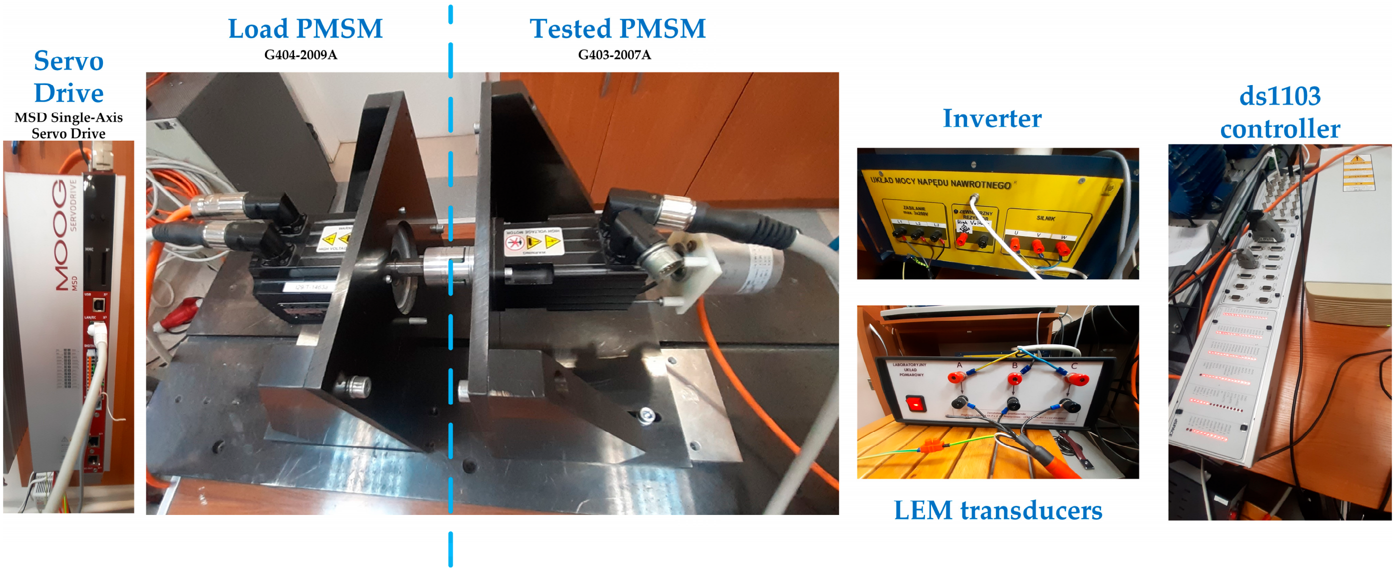


**Figure 15.** Photos of test stand elements.

**Table 5.** Parameters of training and testing vectors in experimental studies.

| Feature | Training Data | Testing Data |
|---|---|---|
| Number of samples | 1,260,162 | 840,096 |
| Speed values | $\pm 0.1\omega_{ref}$, $\pm 0.2\omega_{ref}$, $\pm 0.35\omega_{ref}$ | $\pm 0.08\omega_{ref}$, $\pm 0.25\omega_{ref}$, $\pm 0.4\omega_{ref}$ |
| Load Values | $0.1T_N$, $0.2T_N$ | $0.15T_N$ |

In the case of experimental studies, the average p-value was also determined, which was 0.0093 for the training vector and 0.014 for the test vector. In this case, the obtained data are significant and the null hypothesis is rejected (*p*-value < 0.05). The correlation values of individual elements of the training input vector and the output vector are shown in Figure 16. The inputs from previous samples, in particular the speed error, have the greatest impact on the output value. These values also indicate that the expansion of the input vector should be based on subsequent values of the speed error in the previous samples. The influence of the values from previous samples in the case of current errors is not as significant as in the simulation studies.

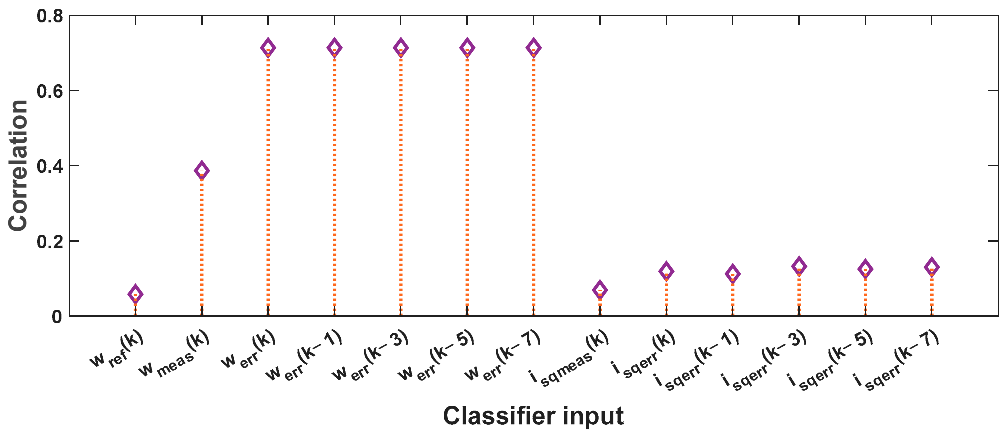


**Figure 16.** Correlation values between output vector and individual input vector elements of training data in experimental studies.

In this case, also, statistical data were presented first using confusion matrices. In the event of experimental research, the advantage of the type II classifier is particularly visible (Figure 17). Encoder pulse loss is detected correctly for test data with a more than 10% higher efficiency. Only the scaling error is detected by the type I classifier without the convolutional layer with greater accuracy (5% higher effectiveness) for test data. This damage, however, does not cause significant symptoms at low speeds and is mistaken for a non-damaged operation. This error could be improved by extending the input vector with additional values of constant speed. In the case of experimental studies, the influence of speed on the effectiveness of damage classification was also analyzed (Figure 18). The results are presented for static motor states for samples with simulated damages. In this case, the basic conclusion is also the increase in effectiveness with increasing speed.

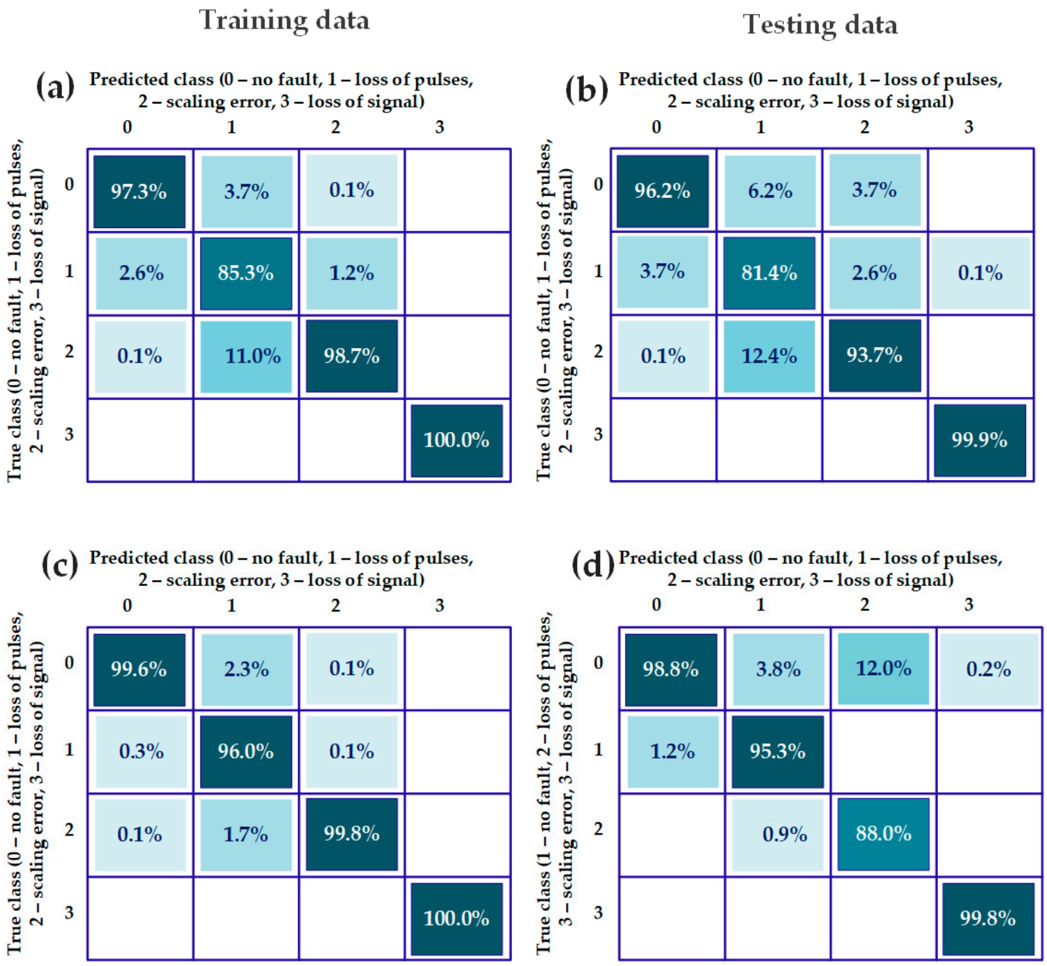


**Figure 17.** Confusion matrices for training and testing data for type I classifier (**a**,**b**) and type II classifier (**c**,**d**) in experimental studies.

Figure 19 shows the waveforms for the total signal loss during the speed in both directions. Classification proceeds with high efficiency. A slightly better fit to the expected value can be observed for the type II classifier. Single errors appear only in dynamic states. Additionally, the activation of a fault in a transient state does not result in its incorrect classification.

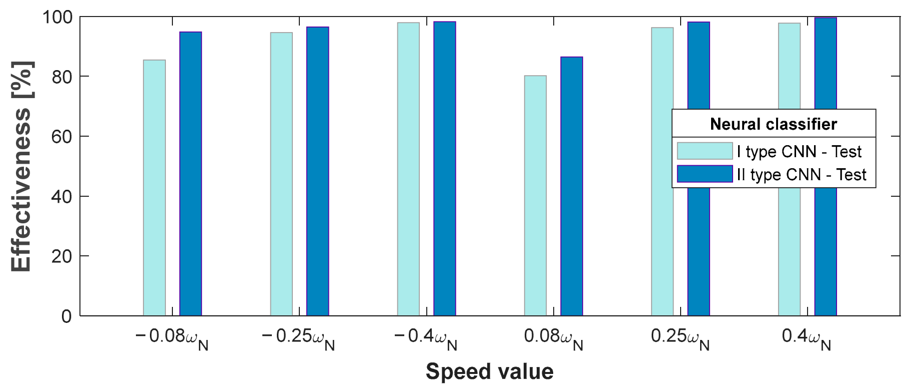


**Figure 18.** Percentage results of classifier effectiveness depending on speed in experimental studies for both types of classifier.

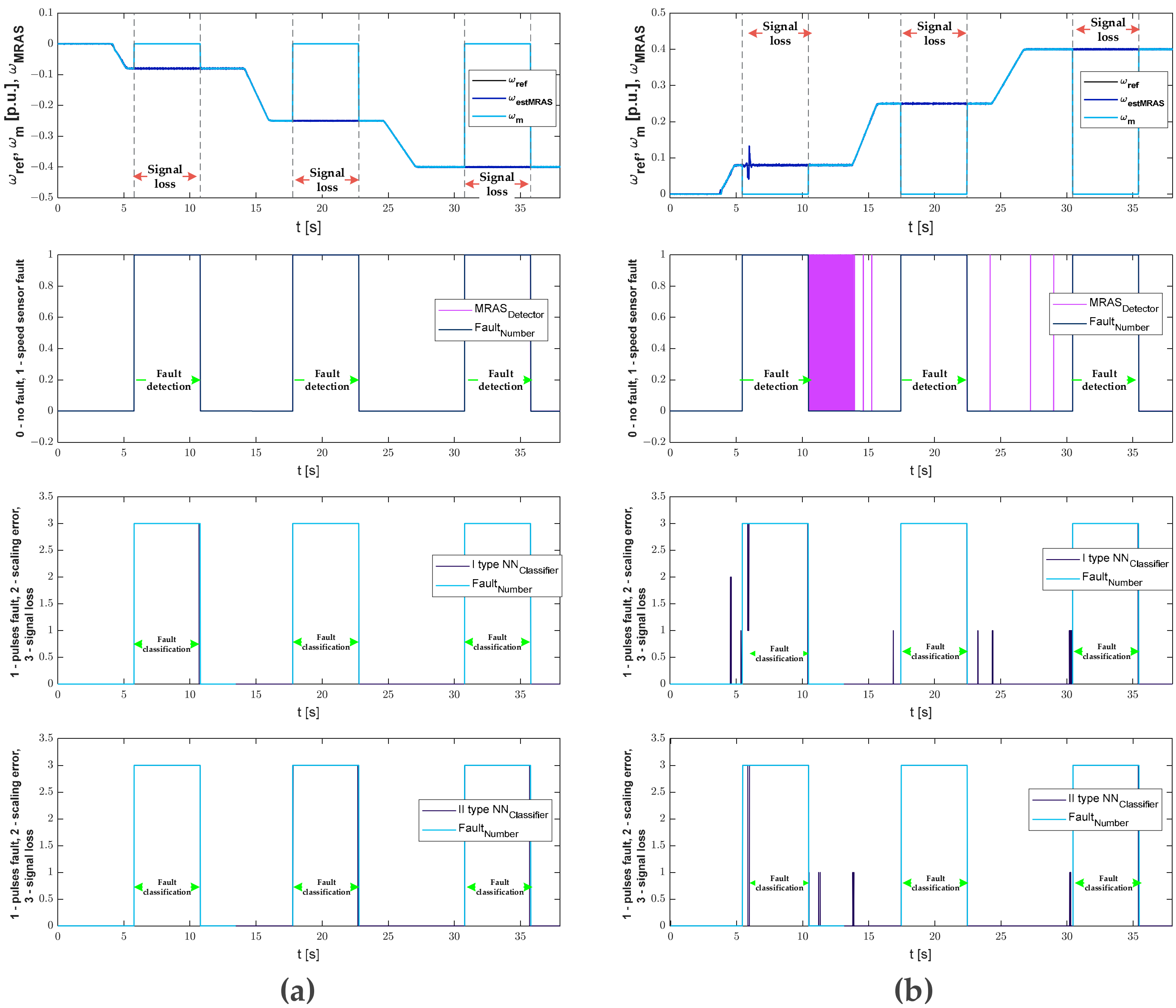


**Figure 19.** Transients speed, MRAS detector, and fault classifiers during the loss of signal fault for negative (**a**) and positive (**b**) speed direction.

Subsequent results are shown for scaling error (error from the start of the measurement—simulation of erroneous indication resulting from the incorrect parameterization of the measurement by the user) and decay of pulses. Based on Figure 20, it can be seen that the scaling error is detected only from about a 0.1 $\omega_{ref}$ value in experimental studies, which is an effect similar to the one found in the simulation results (classification scaling error from about 0.15 $\omega_{ref}$). The results obtained from the encoder pulses loss classification are also close to the simulation results. This damage is mistaken for a scaling error in several samples, and the output of the classifier is not constant.

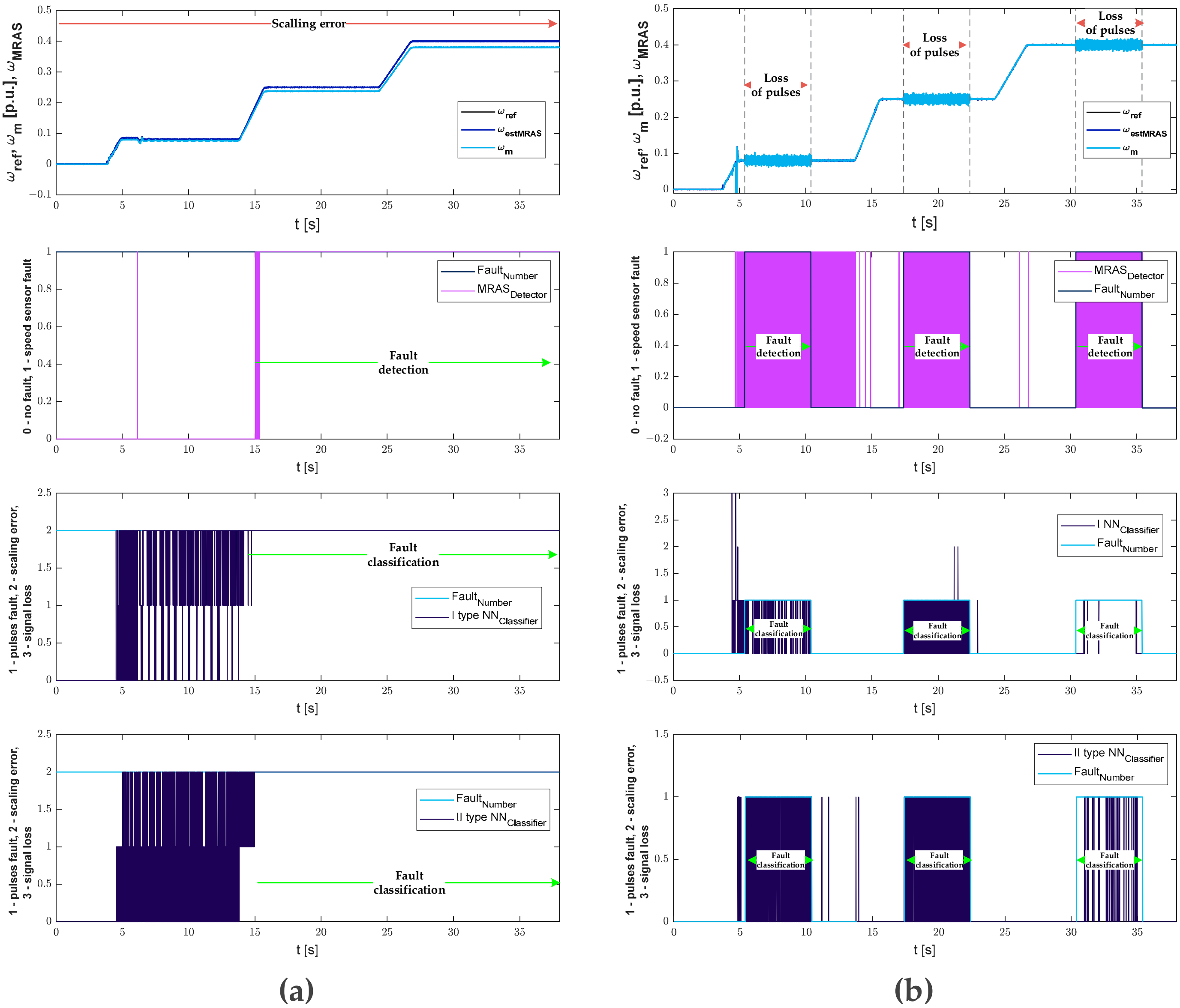


**Figure 20.** Transients of speed, MRAS detector, and fault classifiers during scaling error (**a**) and loss of pulses fault (**b**).

The last element of this section is the presentation of the obtained percentage results of classification based on DNN and shallow neural networks presented in the work [33] (Figure 21). These results confirm that it is worth using more complex neural structures, as they increase the efficiency of classification. The type II classifier is characterized by the highest efficiency. Deep neural networks using a simple MRAS estimator allow for a higher effectiveness than using shallow neural networks with a complex SMO estimator.

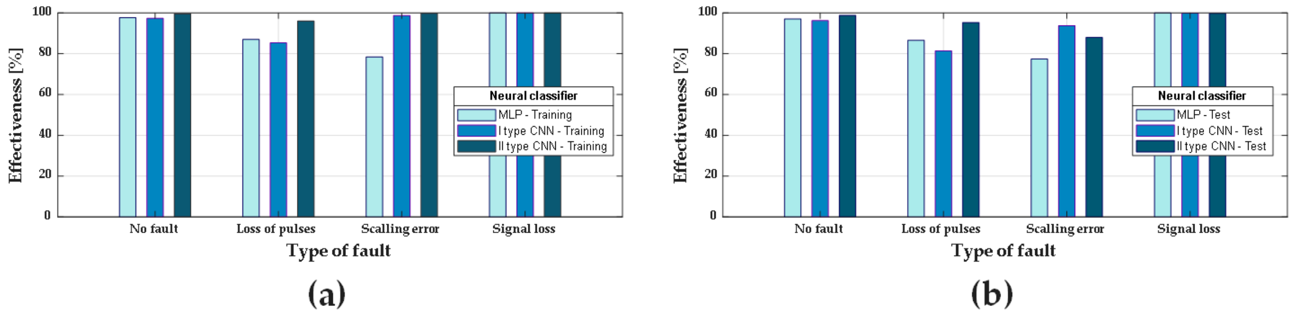


**Figure 21.** Comparison of obtained effectiveness with the use of shallow neural networks (MLP) and DNN with training data (**a**) and testing data (**b**).

## 6. Conclusions

This article presents the possibilities of using deep neural networks in the optoelectronics encoder fault detection and classification system. This article presents solutions with two types of deep neural networks as classifiers and an MRAS speed estimator as fault detector. An important aspect of this work is also the incremental encoder model used to simulate the loss of pulses and information by the sensor. This article presents simulation and experimental results during offline classification. On the basis of the obtained results, the following conclusions can be formulated:

- The simulation of pulse loss using the encoder model confirms that such damage partially causes an effect similar to measurement noise;
- The use of deep neural networks improves the efficiency of damage classification in relation to shallow neural networks—both for training and testing data, for each of the analyzed damage types, the MLP is less effective than any of the DNN classifiers;
- The use of deep neural networks with convolutional layers makes it possible to obtain higher efficiency in classifying optoelectronics encoder faults—a lower result than the classifier without convolutional layers was obtained only for the scaling error classification in experimental studies (lower by 5%);
- Despite the high fit to the training data (nearly 100% Training Accuracy), the classifiers show flexibility in operation on unknown samples;
- Scaling error is the fault most difficult to detect at low speeds, while pulse loss is more difficult to recognize during the entire duration. In addition, as the resolution of the encoder increases, it is less noticeable.

The presented solution also has its limitations. The main one is the sensitivity of the MRAS estimator to motor parameters. For this reason, the authors plan to develop an estimator based on deep neural networks in further research. After appropriate normalization of the input vector, this could increase the universality of the solution. An additional extension may be the development of the damage model. In particular, for the damage due to incorrect scaling and pulse losses.

The process of selecting the layers of the neural structure in the case of using deep neural networks is a complex task and is not subject to more precisely defined rules. Therefore, in further work, the authors plan to conduct research on online speed sensor damage classification and analyze the operation of an extended number of neural structures. Such studies may also be performed using a different type of observer, for example SMO, which made it possible to obtain higher efficiency in previous studies with shallow neural networks.

**Author Contributions:** Conceptualization, M.D. and K.J.; methodology, M.D. and K.J.; software, K.J.; validation, M.D. and K.J.; formal analysis, M.D. and K.J.; investigation, M.D. and K.J.; resources, M.D. and K.J.; data curation, K.J.; writing—original draft preparation, K.J.; writing—review and editing, M.D.; visualization, K.J.; supervision, M.D.; project administration, M.D.; funding acquisition, M.D. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was supported by statutory funds of the Faculty of Electrical Engineering, Wroclaw University of Science and Technology (2023).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

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

## References

1. Chen, Z.; Liang, K.; Peng, T.; Wang, Y. Multi-Condition PMSM Fault Diagnosis Based on Convolutional Neural Network Phase Tracker. *Symmetry* **2022**, *14*, 295. [CrossRef]
2. Laadjal, K.; Bento, F.; Henriques, K.; Cardoso, A.J.M.; Sahraoui, M. A Novel Indicator-Based Online Diagnostics Technique of Interturn Short-Circuit Faults in Synchronous Reluctance Machines. *IEEE J. Emerg. Sel. Top. Power Electron.* **2023**, *11*, 3492–3501. [CrossRef]
3. Assoun, I.; Idkhajine, L.; Nahid-Mobarakeh, B.; Meibody-Tabar, F.; Monmasson, E.; Pacault, N. Wide-Speed Range Sensorless Control of Five-Phase PMSM Drive under Healthy and Open Phase Fault Conditions for Aerospace Applications. *Energies* **2023**, *16*, 279. [CrossRef]
4. Skowron, M.; Orlowska-Kowalska, T.; Kowalski, C.T. Diagnosis of Stator Winding and Permanent Magnet Faults of PMSM Drive Using Shallow Neural Networks. *Electronics* **2023**, *12*, 1068. [CrossRef]
5. Attaianese, C.; D'Arpino, M.; Monaco, M.D.; Di Noia, L.P. Modeling and Detection of Phase Current Sensor Gain Faults in PMSM Drives. *IEEE Access* **2022**, *10*, 80106–80118. [CrossRef]
6. Pietrzak, P.; Wolkiewicz, M.; Orlowska-Kowalska, T. PMSM Stator Winding Fault Detection and Classification Based on Bispectrum Analysis and Convolutional Neural Network. *IEEE Trans. Ind. Electron.* **2023**, *70*, 5192–5202. [CrossRef]
7. Siddiqui, K.M.; Bakhsh, F.I.; Ahmad, R.; Solanki, V. Advanced Signal Processing Based Condition Monitoring of PMSM for Stator-inter Turn Fault. In Proceedings of the 2021 IEEE 8th Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON), Dehradun, India, 8–10 November 2021; pp. 1–4. [CrossRef]
8. Canseven, H.T.; Ünsal, A. Performance Improvement of Fault-Tolerant Control for Dual Three-Phase PMSM Drives Under Inter-Turn Short Circuit Faults. In Proceedings of the IECON 2021—47th Annual Conference of the IEEE Industrial Electronics Society, Toronto, ON, Canada, 20–23 October 2021; pp. 1–5. [CrossRef]
9. Ullah, Z.; Lodhi, B.A.; Hur, J. Detection and Identification of Demagnetization and Bearing Faults in PMSM Using Transfer Learning-Based VGG. *Energies* **2020**, *13*, 3834. [CrossRef]
10. Ahani, H.; Izadi, V. Detection of Different Bearing Faults in a Permanent Magnet Synchronous Motor using PWT and LVQNN. In Proceedings of the 2020 XI National Conference with International Participation (ELECTRONICA), Sofia, Bulgaria, 14–15 May 2020; pp. 1–4. [CrossRef]
11. Ye, M.; Zhang, J.; Yang, J. Bearing Fault Diagnosis under Time-Varying Speed and Load Conditions via Observer-Based Load Torque Analysis. *Energies* **2022**, *15*, 3532. [CrossRef]
12. Gmati, B.; Jlassi, I.; Khojet El Khil, S.; Marques Cardoso, A.J. Open-switch fault diagnosis in voltage source inverters of PMSM drives using predictive current errors and fuzzy logic approach. *IET Power Electron.* **2021**, *14*, 1059–1072. [CrossRef]
13. Chen, T.; Pan, Y.; Xiong, Z. A Hybrid System Model-Based Open-Circuit Fault Diagnosis Method of Three-Phase Voltage-Source Inverters for PMSM Drive Systems. *Electronics* **2020**, *9*, 1251. [CrossRef]
14. Jiang, C.; Liu, H.; Wheeler, P.; Wu, F.; Cai, Z.; Huo, J. A Novel Open-Circuit Fault Detection and Location for Open-End Winding PMSM Based on Differential-Mode Components. *IEEE Trans. Ind. Electron.* **2022**, *69*, 7776–7786. [CrossRef]
15. Zhang, G.; Wang, G.; Wang, G.; Huo, J.; Zhu, L.; Xu, D. Fault Diagnosis Method of Current Sensor for Permanent Magnet Synchronous Motor Drives. In Proceedings of the 2018 International Power Electronics Conference (IPEC-Niigata 2018—ECCE Asia), Niigata, Japan, 20–24 May 2018; pp. 1206–1211.
16. Shen, Q.; Yue, C.; Goh, C.H.; Wang, D. Active Fault-Tolerant Control System Design for Spacecraft Attitude Maneuvers with Actuator Saturation and Faults. *IEEE Trans. Ind. Electron.* **2019**, *66*, 3763–3772. [CrossRef]
17. Bensalem, Y.; Kouzou, A.; Abbassi, R.; Jerbi, H.; Kennel, R.; Abdelrahem, M. Sliding-Mode-Based Current and Speed Sensors Fault Diagnosis for Five-Phase PMSM. *Energies* **2022**, *15*, 71. [CrossRef]
18. Hezzi, A.; Abdelkrim, M.N.; Elghali, S.B. Robust Active Fault Tolerant Control for Five-Phase PMSM against Speed Sensor Failures. In Proceedings of the 2021 18th International Multi-Conference on Systems, Signals & Devices (SSD), Monastir, Tunisia, 22–25 March 2021; pp. 575–579. [CrossRef]
19. Bouakoura, M.; Naăźt-Saăźd, N.; Nait-Said, M.-S. Speed sensor faults diag-348 nosis in an induction motor vector controlled drive. *Acta Electrotech. Et Inform.* **2017**, *17*, 49–51. [CrossRef]
20. Kommuri, S.K.; Defoort, M.; Karimi, H.R.; Veluvolu, K.C. A Robust Observer-Based Sensor Fault-Tolerant Control for PMSM in Electric Vehicles. *IEEE Trans. Ind. Electron.* **2016**, *63*, 7671–7681. [CrossRef]

21. Ben Slimen, S.; Bourogaoui, M.; Sethom, H.B.A. A New Approach for Effective Position/Speed Sensor Fault Detection in PMSM Drives. In *Electrimacs Lecture Notes in Electrical Engineering*; Zamboni, W., Petrone, G., Eds.; Springer: Cham, Switzerland, 2019; Volume 615.
22. Zhao, H.; Luo, P.; Wang, N.; Zheng, Z.; Wang, Y. Fuzzy logic control of the fault-tolerant PMSM servo system based on MRAS observer. In Proceedings of the 2018 Chinese Control and Decision Conference (CCDC), Shenyang, China, 9–11 June 2018; pp. 1812–1817.
23. Bourogaoui, M.; Jlassi, I.; El Khil, S.K.; Sethom, H.B.A. An effective encoder fault detection in PMSM drives at different speed ranges. In Proceedings of the 2015 IEEE 10th International Symposium on Diagnostics for Electrical Machines, Power Electronics and Drives (SDEMPED), Guarda, Portugal, 1–4 September 2015; pp. 90–96.
24. Maulik, S.; Konar, P.; Chattopadhyay, P. Stator Current Based Multi-Class Fault Diagnosis of Three Phase Induction Motor using Machine Learning Framework. In Proceedings of the 2022 IEEE 6th International Conference on Condition Assessment Techniques in Electrical Systems (CATCON), Durgapur, India, 17–19 December 2022; pp. 225–229. [CrossRef]
25. Skowron, M.; Orlowska-Kowalska, T.; Kowalski, C.T. Detection of Permanent Magnet Damage of PMSM Drive Based on Direct Analysis of the Stator Phase Currents Using Convolutional Neural Network. *IEEE Trans. Ind. Electron.* **2022**, *69*, 13665–13675. [CrossRef]
26. Jung, H.; Choi, S.; Lee, B. Rotor Fault Diagnosis Method Using CNN-Based Transfer Learning with 2D Sound Spectrogram Analysis. *Electronics* **2023**, *12*, 480. [CrossRef]
27. Kim, M.; Jung, J.H.; Ko, J.U.; Kong, H.B.; Lee, J.; Youn, B.D. Direct Connection-Based Convolutional Neural Network (DC-CNN) for Fault Diagnosis of Rotor Systems. *IEEE Access* **2020**, *8*, 172043–172056. [CrossRef]
28. Eren, L.; Ince, T.; Kiranyaz, S. A Generic Intelligent Bearing Fault Diagnosis System Using Compact Adaptive 1D CNN Classifier. *J. Sign. Process Syst.* **2019**, *91*, 179–189. [CrossRef]
29. Pham, M.-T.; Kim, J.-M.; Kim, C.-H. 2D CNN-Based Multi-Output Diagnosis for Compound Bearing Faults under Variable Rotational Speeds. *Machines* **2021**, *9*, 199. [CrossRef]
30. Dybkowski, M.; Klimkowski, K. Artificial Neural Network Application for Current Sensors Fault Detection in the Vector Controlled Induction Motor Drive. *Sensors* **2019**, *19*, 571. [CrossRef] [PubMed]
31. Skowron, M.; Teler, K.; Adamczyk, M.; Orlowska-Kowalska, T. Classification of Single Current Sensor Failures in Fault-Tolerant Induction Motor Drive Using Neural Network Approach. *Energies* **2022**, *15*, 6646. [CrossRef]
32. Jankowska, K.; Dybkowski, M. Experimental Analysis of the Current Sensor Fault Detection Mechanism Based on Neural Networks in the PMSM Drive System. *Electronics* **2023**, *12*, 1170. [CrossRef]
33. Jankowska, K.; Dybkowski, M.; Petro, V.; Kyslan, K. Classification of Speed Sensor Faults Based on Shallow Neural Networks. *Appl. Sci.* **2023**, *13*, 7263. [CrossRef]
34. Klimkowski, K.; Dybkowski, M. Concept of the speed sensor faults detector for DFOC drive based on the neural network. *Comput. Appl. Electr. Eng.* **2015**, *13*, 340–351.
35. Incze, I.I.; Negrea, A.C.; Imecs, M.; Szabo, C. Incremental encoder based position and speed identification: Modeling and simulation. *Acta Univ. Sapientiae Elect. Mech. Eng.* **2010**, *2*, 27–39.
36. Incze, J.J.; Szabó, C.; Imecs, M. Modeling and simulation of an incremental encoder used in electrical drives. In Proceedings of the 10th International Symposium of Hungarian Researchers on Computational Intelligence and Informatics, Budapest, Hungary, 25–26 March 2009; pp. 97–109.
37. Kevin, M.; Lynch, N.M.; Matthew, E.L. *Chapter 21—Sensors, Embedded Computing and Mechatronics with the PIC32*; Newnes: Oxford, UK, 2016; pp. 317–340, ISBN 9780124201651. [CrossRef]
38. Pop, A.A. Incremental Encoder Speed Acquisition Using an STM32 Microcontroller and NI ELVIS. *Sensors* **2022**, *22*, 5127. [CrossRef]
39. Skowron, M.; Orlowska-Kowalska, T.; Wolkiewicz, M.; Kowalski, C.T. Convolutional Neural Network-Based Stator Current Data-Driven Incipient Stator Fault Diagnosis of Inverter-Fed Induction Motor. *Energies* **2020**, *13*, 1475. [CrossRef]