Research Article

Artificial Neural Network-Based Control of Switched Reluctance Motor for Torque Ripple Reduction

Iqra Tariq [iD],[1] **Raheel Muzzammel** [iD],[1] **Umar Alqasmi,**[2] **and Ali Raza** [iD][1]

[1]*Department of Electrical Engineering, University of Lahore, Lahore, Pakistan*
[2]*Department of Electrical and Computer Engineering, King Abdulaziz University, Jeddah, Saudi Arabia*

Correspondence should be addressed to Iqra Tariq; ekrahtariq@gmail.com

Switched reluctance motor is acquiring major attention because of its simple design, economic development, and reduced dependability. These attributes make switched reluctance motors superior to other variable speed machines. The major challenge associated with the development of a switched reluctance motor is its high torque ripple. Torque ripple produces noise and vibration, resulting in degradation of its performance. Various techniques are developed to cope with torque ripples. Practically, there exists not a single mature technique for the minimization of torque ripples in switched reluctance motors. In this research, a switched reluctance motor is modelled and analysed. Its speed and current control are implemented through artificial neural networks. Artificial neural network is found to be a promising technique as compared with other techniques because of its accuracy, reduced complexity, stability, and generalization. The Levenberg–Marquardt algorithm is utilized in artificial neural networks due to its fast and stable convergence for training and testing. It is found from research that artificial neural network-based improved control shows better performance of the switched reluctance motor. Realization of this technique is further validated from its mean square error analysis. Operating parameters of the switched reluctance motor are improved significantly. Simulation environment is created in Matlab/Simulink.

1. Introduction

Switched reluctance motor (SRM) is a type of stepper motor, an electric motor that runs by reluctance torque. Unlike common DC motors, power is delivered to the windings in the stator rather than the rotor. No magnets or coils are attached to the rotor. SRM consists of a position sensor that senses the energized pole of a stator and then aligns the rotor with the energized stator pole [1–3]. Because of their exceptional characteristics, SRM has great potential in motion control applications, which allows good performance in harsh dusty environments with high-temperature degrees [4–6].

SRM is an attractive alternative to AC and DC motors for general-purpose industrial drives, as well as for high-performance automotive drives and other applications because of its simple structure due to the absence of magnets, rotor conductors, and brushes. It is used instead of induction and DC motors in many applications because of its low construction cost, high reliability, high power density, fast dynamic response, good controllability ruggedness, and fault tolerance [6–8].

In comparison with conventional machines, the primary disadvantage of SRM is higher torque ripple, which contributes to acoustic noise and vibration. The origin of torque pulsations is due to the nonlinear and discrete nature of the torque production mechanism. Minimization of torque ripple is essential for high performance [9]. To optimize the design of SRM, magnetic characteristics play an important role. The performance of SRM is dependent on its design. Torque ripples of SRM can be reduced by machine design or by their control circuits. Many researchers used finite element programs to design motors and obtained its torque, current, and rotor angle $(\tau - i - \theta)$ characteristics. Obtained data stored in 3D lookup table were being used to investigate the corresponding suitable current that leads to minimum

torque, but this method consumes much time [6, 10]. Analytical methods were implemented for torque ripple reduction but these methods lack modelling and computational accuracy. To achieve minimized reluctance, ripples are required to be reduced as torque and speed are inversely proportional to each other [11]. Direct instantaneous torque control (DITC) technique was utilized for the reduction of torque ripple, but it was limited in operation range, and the controller needs to be changed as speed increases [12]. The multiphase torque-sharing scheme was proposed to control the torque ripple, but it adds complexity to the system [13]. TSF (torque sharing function) and current control methods were employed for torque ripple minimization but these methods neglect mutual torque that has a significant effect in certain applications [14]. A unified controller for switched reluctance motors was proposed for wide speed operations. The proposed controller achieves minimum output torque ripple at low and medium speed and operates in the single pulse mode at high speed [15]. In [16], by applying the method of direct torque control (DTC), torque ripple and flux are regulated within the hysteresis limit.

The performance of SRM is improved by the advanced artificial intelligence techniques such as fuzzy logic and artificial neural network (ANN), or their combinations can be implemented [17]. The artificial neural network (ANN) method is inspired by the human brain model and does not need a large memory to store the magnetic characteristics. This technique has many benefits such as fault tolerance, organic learning, linear data processing, and self-repair ability. It can perform in noisy environments [14]. SRM nonlinearity characteristics are trained by neural networks and then the current graph for ripple reduction is obtained. ANN is used as an intelligent controller [18, 19]. Torque ripple reduction is done through PI and fuzzy logic controller [20]. SRM torque ripples are reduced in four quadrants of operation through a control scheme, which is an extension of the TSF technique. For the calculation of static characteristics of the motor, a 2D finite element program is used. Torque command is used for the estimation of a current of motor, and ANN is used for calculation of the angle of the rotor [21]. In [22], ANN is used for direct torque control of four-phase SRM for minimization of torque ripples and to overcome the drawback of voltage space vector in classical DTC technique.

In this research, an ANN-based algorithm is developed and is applied on SRM for torque ripple reduction. It is found from simulations that ANN is an accurate and less complex algorithm and offers performance improvement in the dynamic environment particularly in the case of SRM. The proposed control technique computes the desired output with accuracy and offers speedily conversions in less computational time in comparison with the PI controller and simplified model. This research includes simulation of different cases of the proposed model. Torque ripple value and torque improvement percentage are evaluated in different cases of the proposed model. Torque is improved by 1.44 times as compared with the simplified model. Reduction in torque ripple and improvement in torque increase the SRM speed. The proposed technique has potential to improve industrial acceptance of SRM. This potential is validated by RMSE analysis.

The rest of the paper is organized as follows: In Section 2, it explains the mathematical formulation of SRM, ANN, and Levenberg–Marquardt algorithm. Section 3 illustrates the control scheme of ANN-based SRM, and the basic process is explained. Implementation of the proposed scheme is described in Section 4. Simulation results and discussions will be given in Section 5. Section 6 and Section 7 show the effectiveness of the proposed technique and conclusion, respectively.

2. Mathematical Formulation

2.1. Mathematical Modelling of SRM. SRM is a rotating electric machine in which salient poles are possessed by the rotor and stator. Therefore, a machine is referred to as a doubly salient machine. It comprises of a stator with an exciting winding and magnetic rotor. No permanent magnet is required because the tendency of rotor poles to align with excited poles, so as to minimize the stator flux linkages that result from a given applied stator current, is the source of torque production. Generally, the motor phase equations describe the electrical behaviour of SRM. The instantaneous voltage across the terminals of a phase of SRM winding is related to the flux linked in the winding which is obtained by Faraday's law. Due to double salient construction and magnetic saturation effects, flux linked in the SRM phase varies as a function of rotor position θ and phase current i:

$$V_j = RI_j + \frac{\partial \varphi_j(i,\theta)}{\partial i}\frac{\mathrm{d}i}{\mathrm{d}t} + \frac{\partial \varphi_j(i,\theta)}{\partial \theta}\frac{\mathrm{d}\theta}{\mathrm{d}t}, \quad \text{here } j = 1,2,3,$$

$$\omega = \frac{\partial \theta}{\partial t}. \tag{1}$$

Flux in each phase is given after excluding saturation and mutual inductance effects:

$$\varphi_j(\theta, i_j) = L(\theta)i_j, \tag{2}$$

which is written as follows:

$$V_j = RI_j + L(\theta)\frac{\partial i}{\partial t} + i\frac{\partial L(\theta)}{\partial \theta}\omega. \tag{3}$$

Energy associated with three phases ($n = 3$) is given as follows:

$$W_{\text{total}} = \frac{1}{2}\sum_{j=1}^{3} L(\theta + (n-j-1)\theta_s)I_j^2. \tag{4}$$

Phase inductance displaced by angle θ_s and torque of motor is as follows:

$$T_e = \frac{\partial W_{\text{total}}}{\partial \theta} = \frac{1}{2}\sum_{j=1}^{3}\frac{\partial L(\theta + (n-j-1)\theta_s)}{\partial \theta}I_j^2. \tag{5}$$

The mechanical equation which describes the mechanical motion of the motor is written as follows:

$$J \frac{\partial \omega}{\partial t} = T_e - T_l - f\omega, \quad (6)$$

where V is the terminal voltage, I is a phase current, R is the phase winding resistance, $\varnothing$ is the flux linked by winding, J is a moment of inertia, f is the friction coefficient, (θ) is the instantaneous inductance, T_l is the torque load, and T_e is the total torque.

2.2. Mathematical Steps of ANN.

Artificial neural networks can learn and model nonlinear and complex relationships. ANN can simplify after learning from the initial inputs and it can infer unseen relationships on unseen data as well, thus making the model generalize and predict on unseen data. The general equation of ANN is as follows:

$$Y = \sum (\text{weight} * \text{input}) + \text{bias}. \quad (7)$$

A node is a basic unit of a neural network, which gives a certain number of inputs and bias values. When the signal (value) arrives, it is multiplied by the weight value. Every input has its weight value which can be tuned during the training stage. Bias is an extra input to neuron, and it is always one and it has its connection weight [23, 24]:

$$z = x_1 * w_1 + x_2 * w_2 +, \ldots, +x_n * w_n + b * 1. \quad (8)$$

Outputs and inputs in neural networks are in linear form i.e., 0 and 1. Activation function introduces nonlinearity. Sigmoid or logistic activation function is mostly used for binary classification problems (output values that ranges 0-1). The calculation of the derivative of the sigmoid function is easy. Then, to change the linear values into nonlinear, sigmoid function is used [23]:

$$\widehat{Y} = a_{\text{out}} = \text{sigmoid}(z), \quad (9)$$

where $\widehat{Y}$ is the output and sigmoid is the activation function applied to bias and a weighted sum of inputs [24].

2.3. Mathematical Operations of the Levenberg–Marquardt Algorithm.

LMA provides a numerical solution to the minimization problem of a nonlinear function. It is fast and has a stable convergence. This algorithm is appropriate for training small- and medium-sized problems in the artificial neural networks field. LMA is the combination of the steepest descent method and the Gauss–Newton algorithm. It inherits the stability of the steepest descent method and speed advantage of the Gauss–Newton algorithm [25]. The Hessian matrix can be approximated as follows:

$$H = J^T J. \quad (10)$$

A gradient can be computed as follows:

$$g = J^T e. \quad (11)$$

LMA uses an approximation to the Hessian matrix in the following Newton-like update [24]:

$$X_{K+1} = X_k - \left[J^T J + \mu I \right]^{-1} J^T e, \quad (12)$$

where J is the Jacobin matrix that contains first derivatives of network errors with respect to weights and biases, e is a vector of network errors, I is the identity matrix, and μ is positive, called combination coefficient. The LMA switches between the two algorithms during the training process. When the combination coefficient μ is very small (nearly zero), the Gauss–Newton algorithm is used [25].

The update rule of the Gauss–Newton algorithm is written as follows:

$$X_{K+1} = X_k - \left(J^T J \right)^{-1} J^T e. \quad (13)$$

When a combination coefficient μ is very large, the steepest descent method is used. The update rule of the steepest descent algorithm is written as follows:

$$X_{K+1} = X_k - \alpha g_k. \quad (14)$$

where α is the learning constant. If μ is very big, it can be represented as the learning coefficient in the steepest descent method:

$$\alpha = \frac{1}{\mu}. \quad (15)$$

3. Proposed Methodology

Torque ripple affects the speed performance of switched reluctance motor (SRM), and to overcome this problem, artificial neural network (ANN)-based modelling of SRM is done. The ANN-based control scheme of the switched reluctance motor is presented in Figure 1. SRM cannot start directly from DC supply; it needs a converter for its operation. The SRM model is based on a 3-phase asymmetric power converter which consists of three legs. Each leg consists of two IGBTs and two freewheeling diodes. SRM winding is connected to positive voltage source through IGBTs in the active mode. During the conduction period, positive voltages flow in phase windings and vice versa in the nonconduction period. Stored energy returns to DC source through diodes [26]. Mainly, the converter controls the speed of the motor through proper excitation of corresponding windings in the stator. Gate pulses through the hysteresis band are given as an input to the power converter. IGBT switching frequency is determined by the hysteresis band. Reference current is taken for hysteresis control of 3-phase current. The position sensor determines turning off and on phases of motor windings. Noise is added with actual motor speed by using the sum block and given as an input to the ANN-based speed block, and its output is connected with speed of the position sensor. Similarly, noise is added individually with actual 3-phase current of the motor and then 3-phase current given as input to ANN-based 3-phase current blocks and its output to the sum block before the hysteresis band in Figure 1. Three hysteresis controllers generate IGBT drive signals by comparison with ANN-based 3-phase current with references and are used for separately
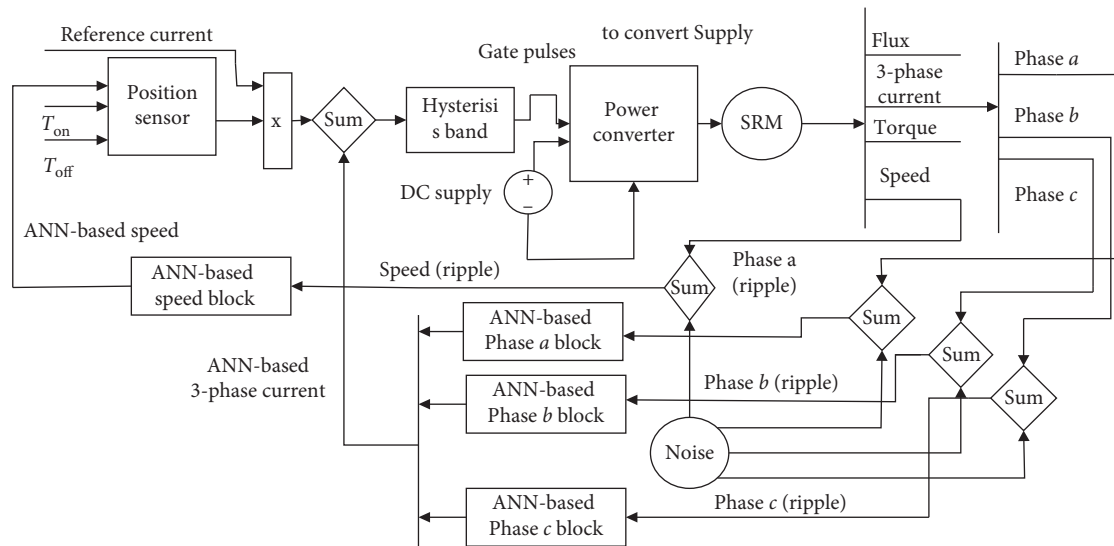
Figure 1: Control scheme of SRM-based ANN by regulating its speed and current.

controlling phase currents. The desired output is obtained by application of ANN-based SRM.

Complex relationships between input and output are found through ANN, which is considered a nonlinear statistical data-modelling tool according to the steps discussed in Figure 2. The first step involved is the normalization of data. The transformation of data into a specific range is known as data normalization. In ANN, input data are normalized, otherwise the network will be ill-conditioned. It is not possible to achieve the same range of values for each of the input to the ANN mode. It assures stable convergence of weights and bias. The second step is the data partition. Random data division (dividerand) is used in ANN training to use maximum data for training (in general splitting up of data) for the ANN model development into training data, validation data, and test data. During the training process, the back-propagation algorithm is used to define the weights on connections and then used for calculating the outputs. Generally, for some applications, these weights can be used to initialize the neural network and then updated by using an online training algorithm. Network weights and biases are updated during training. Validation is used to measure network generalization, and when generalization stops improving, it stops training. Independent measure of network performance during and after training is achieved by testing data and has no effect on training. LMA is used as an online training algorithm. LMA provides a numerical solution to the minimization problem of a nonlinear function. In the artificial neural networks field, for training small- and medium-sized problems, LMA is the best option.

LMA is the combination of the steepest descent method and the Gauss–Newton algorithm. It acquires the speed advantage of the Gauss–Newton algorithm and stability of the steepest descent method. In many cases, it can converge well even if the error surface is much more complex than the quadratic situation, and that is why it is instantaneous than the Gauss–Newton algorithm. In convergent situations, the LMA tends to be a little slower than the Gauss–Newton

algorithm but it converges much faster than the steepest descent method. The basic idea of the LMA is that it executes a combined training process: around the area with complex curvature, the LMA switches to the steepest descent algorithm until the local curvature is right to make a quadratic approximation and then it nearly becomes the Gauss–Newton algorithm, which can speed up the convergence significantly. Weights and biases are updated during training, and data are presented according to which network is adjusted according to its error.

Independent measure of network performance during and after training is achieved by testing data and has no effect on training. The third step is the architecture of the network in which a two-layer feed-forward network is applied in standard function fitting, which comprises of the sigmoid transfer function in the hidden layer and a linear transfer function in the output layer. The fourth step is the learning algorithm used for training of network to fit the inputs and targets. It helps in achieving accurate results and analysis.

The fifth step is the evaluation of network which means that we can test our network on more data and retrain it if we are not satisfied with our obtained results. The sixth step is to determine the deployable solution; in this way, a trained neural network is generated in the form of the Simulink diagram or the form of code. In this research, this algorithm is implemented because of the easy model building and less formal statistical knowledge required. Unlike other prediction techniques, ANN does not impose any restrictions (like how it should be distributed) and it gives data with nonconstant difference and with high volatility. With the evolving technology of ANN, motor fault detection problems can easily be solved using an advanced approach based on handy measurement without the need for expensive equipment and precise mathematical models that are obtained from conventional fault detection techniques. Therefore, it is a better feasible option than any other conventional technique.
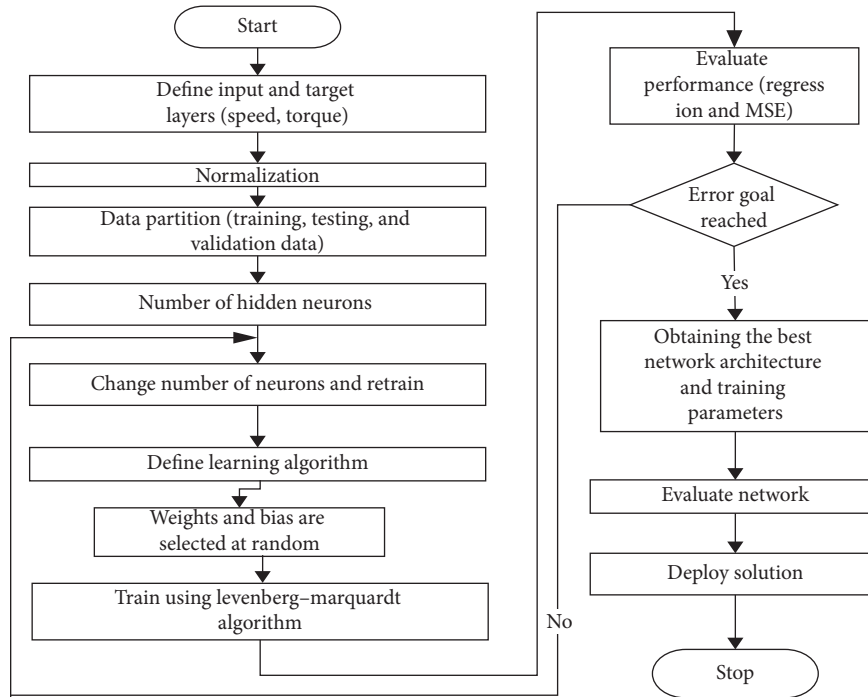
FIGURE 2: Flow chart of artificial neural network (ANN).

4. Implementation

The SRM closed-loop control consists of an outer speed loop and an inner current loop as shown in Figure 3. The position sensor is used to sense the position of the exciting pole. Rotor position θ is sensed by the position sensor. Its derivative value gives rotor speed ω that is compared with the reference value of speed ω_{ref}, and it gives error, which is processed through PI control or fuzzy logic control to yield the reference current I_{ref}. SRM has a sensor that senses the actual current I_{act} of the motor. The reference current I_{ref} is compared with actual current I_{act}, and the error signal goes to the current controller, and these errors are used to determine the switching of a phase of SRM. Then, based on position data obtained from the position sensor, voltages are applied to respective windings. In this way, speed and current are controlled.

SRM magnetic behaviour is highly nonlinear, and there is a fast change in its parameters. PI control does not apply to the systems because it demands a change in PI constants in time. The fuzzy logic controller can handle the nonlinearity and more robust rather than the PI controller. The fuzzy Logic controller has a significant steady-state error and takes much more computing time than for the PI controller [20]. Combination of the PI controller and fuzzy logic controller does not have steady-state error [20]. Three main steps are involved in the implementation of artificial neural network control. Training data of speed and 3-phase current are in rad/s and Ampere (A), respectively. Step one includes the data collection and preprocessing of data. After data collection from a simulation of SRM, the preprocessing of data was done to train ANN more efficiently. It includes normalization of data i.e., 500,001 values of speed and 3-phase

current with respect to time. As the data collected from the simulation were too large that is why mean values were computed after every 10,000 values of speed and 3-phase current. In this way, 51 values of speed and 3-phase current were achieved. It formed the target layer (without ripples data). To design the input layer, applied similar steps for speed and 3-phase current (with ripples data). Step two was to build the ANN model. Random data division was applied to use maximum data for training. This was the usual data split for ANN model development. Different combinations of data divisions were tried, but the data divided into training, validation, and testing in a ratio of 70 : 15 : 15 gave better results. The number of neurons in the hidden layer was given value 10. The Levenberg–Marquardt Algorithm (LMA), which has fast and stable convergence and generally allows importing, creating, using, and exporting neural network data, was used to train the network. Mean square error (MSE) and regression (R) for training, validation, and testing were calculated. After that, retraining was conducted several times by changing the number of hidden layers until the following conditions were fulfilled higher regression value and lower MSE value in training, validation, and testing. The network is evaluated, and the Simulink diagram is generated for a trained neural network. Step three was the simulation of the model. Simulation is an important step to check that a trained neural network-fitting block can generalize well and produce desired outputs. In this control scheme, torque ripple reduction is done by regulating the speed and current through ANN, and for this purpose, trained neural network-fitting blocks of speed and three-phase currents were used in a simulation model of 6/4 (6 rotor poles and 4 stator poles) switched reluctance motor (SRM).
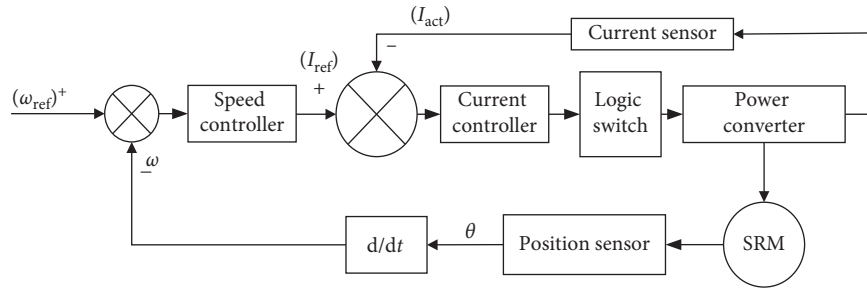
FIGURE 3: Closed loop of SRM.

5. Simulation Results and Discussions

5.1. Simplified Model. Different configurations of SRM such as 12/8, 8/6, 6/4, and 4/2 are available but our focus is on 6/4 configuration. The three-phase asymmetrical power converter was employed to feed SRM. The converter has three legs, each of which consists of two freewheeling diodes and two insulated gate bipolar transistors (IGBTs). IGBTs are used in this converter because of their good current ratings. To drive positive currents into phase windings, active IGBTs apply positive source voltages to stator windings during the conduction cycle. During the freewheeling cycle, a negative voltage is applied to the windings, and the stored energy is returned to power DC source through diodes. The position sensor reduces the fall time of currents in motor windings and controls the turn-on and turn-off angles of motor phases. Three hysteresis controllers generate IGBT drive signals by comparison of measured current with references and are used for separately controlling phase currents. The hysteresis band helps in determining IGBT switching frequency.

For 6/4 SRM, the simulation environment is built in Matlab/Simulink. The DC supply voltage of 240 V is used. Converter turn-on angle = 45° and turn-off angle = 75° are kept constant, respectively, over the speed ranges. Practical range of turn-on and turn-off angles depends on inductance profile and therefore on configuration and pole geometry of particular SRM. Torque waveforms are under control of turning off and on angle of phase windings. These two switching angles play an important role in deciding whether SRM develops positive or negative and high or low electromagnetic torque [27, 28]. In [29], optimal turn on and off (switching angles) were detected as a function of phase current and rotor speed to maximize the electromagnetic torque or minimize the torque ripples [30]. The optimal turn-on and turn-off angles can be determined online to achieve balance between efficiency and torque ripple criteria. The reference current is 200 A, and the hysteresis band is chosen at ±10 A. By applying step reference to regular input, the motor is started. The motor will speed up according to load characteristics. Only currents are controlled, and the motor speed increases according to the mechanical dynamics of the system. Therefore, mechanical and dynamics mainly decide the speeding up of the motor. Basically, the motor has two operating modes, and these modes are dependent upon speed value. From stand still

TABLE 1: SRM Parameters used in Simulink.

Motor parameters	Values
Rated power	60 kW
Number of phases	3
Number of stator poles	6
Number of rotor poles	4
Aligned phase inductance	23.6 mH
Unaligned phase inductance	0.67 mH
Inertia	0.05 kg·m^2
Friction coefficient	0.02 Nms
Stator resistance	0.05 Ω
DC supply voltage	240 V
Maximum current	450 A
Maximum flux linkage	0.486 Vs

up to 3000 rpm, the motor operates in a current control mode. Current is regulated by the reference value of current. Afterwards, for speed above 3000 rpm, the motor operates in the voltage fed mode. In this mode, current cannot be regulated by the reference value of current, and there is no switching because of high emf of the motor. During the active or conduction period, switches remain closed and DC supply directly feeds the motor in the voltage fed mode. In this mode, the average value of developed torque has inverse relationship with motor speed. By simulating the model for 0.5 seconds according to the parameters discussed in Table 1, SRM drive waveforms of magnetic flux, speed, winding current (phase *a*, phase *b*, and phase *c*), and motor torque are displayed on the scope.

The simplified model torque waveform is shown in Figure 4. SRM has high torque ripple due to transitions of current from one phase to the following one. Torque ripple is a specific characteristic of SRM, and the converter's turn-on and turn-off angles are primarily responsible for it. The simplified model has further two cases in which a band-limited white noise block is employed as discussed below.

5.1.1. Simplified Model Case 1. In Figure 5, noise is added with the waveform of the simplified model of SRM and considering the current simple as in the simplified model by setting the value of noise power = 0.0005 W·s and simulating the model for 0.5 seconds according to the parameters discussed in Table 1.

Figure 6 indicates values of speed at $t = 0.1$ s, 0.25 s, and 0.4 s are 4189 rad/s, 5192 rad/s, and 5735 rad/s.
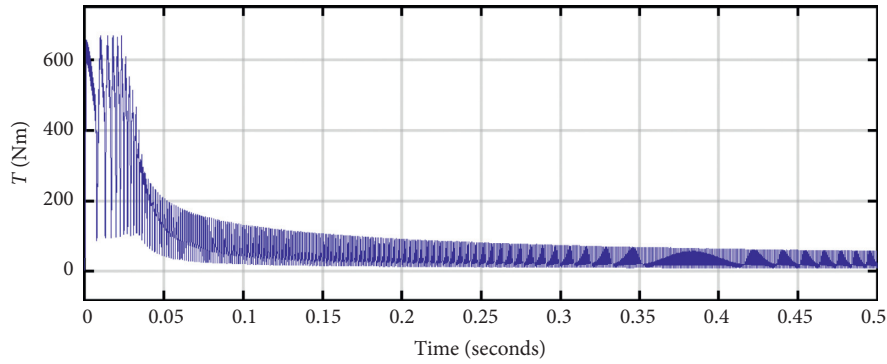
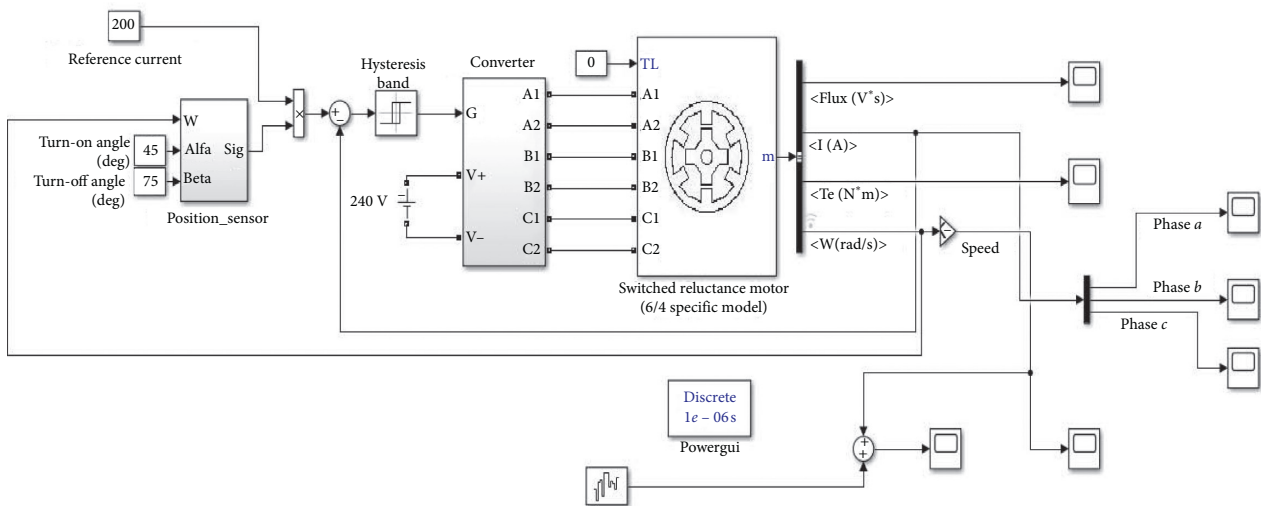FIGURE 4: Simplified model torque waveform.

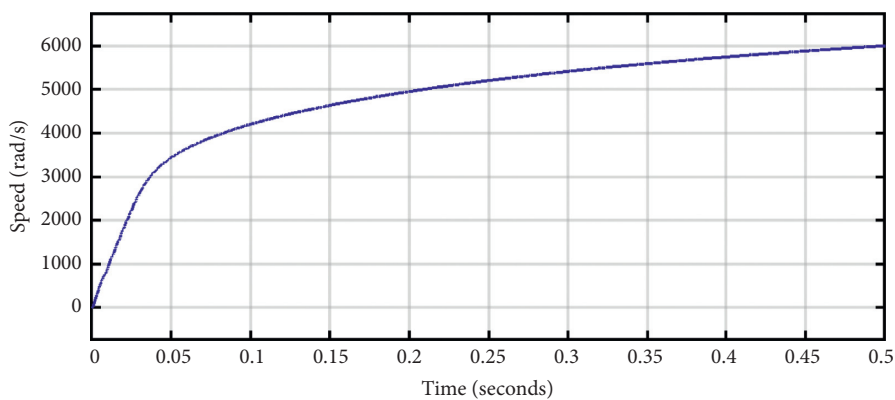FIGURE 5: Simplified model of SRM by adding noise with speed.

FIGURE 6: Speed waveform of the simplified model.

Figure 7 exhibits ripples in waveform of speed and values of speed at time points 0.1 s = 4229 rad/s, at 0.25 s = 5175 rad/s, and at 0.4 s = 5760 rad/s. The significant value of MSE is noticed in the graphical results of Figure 6, and speed with noise waveform is 2.275e + 07%.

5.1.2. Simplified Model Case 2. Figure 8 shows the response of noise with a 3-phase current of SRM. Noise is added with phase a, phase b, and phase c of the simplified model for the same time period and simulation time as in the previous case.
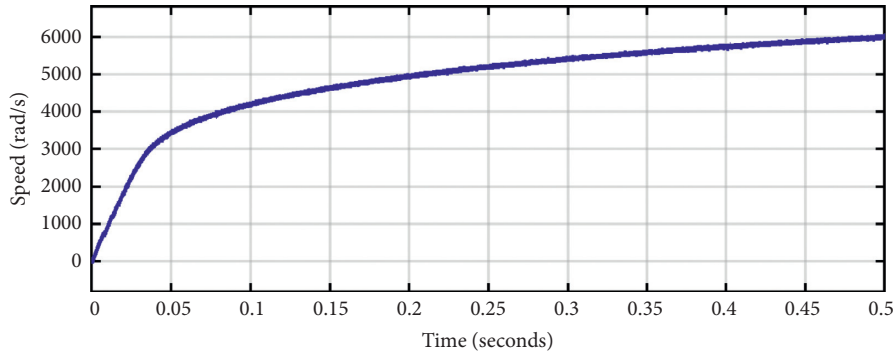
FIGURE 7: Simplified model speed with noise waveform.
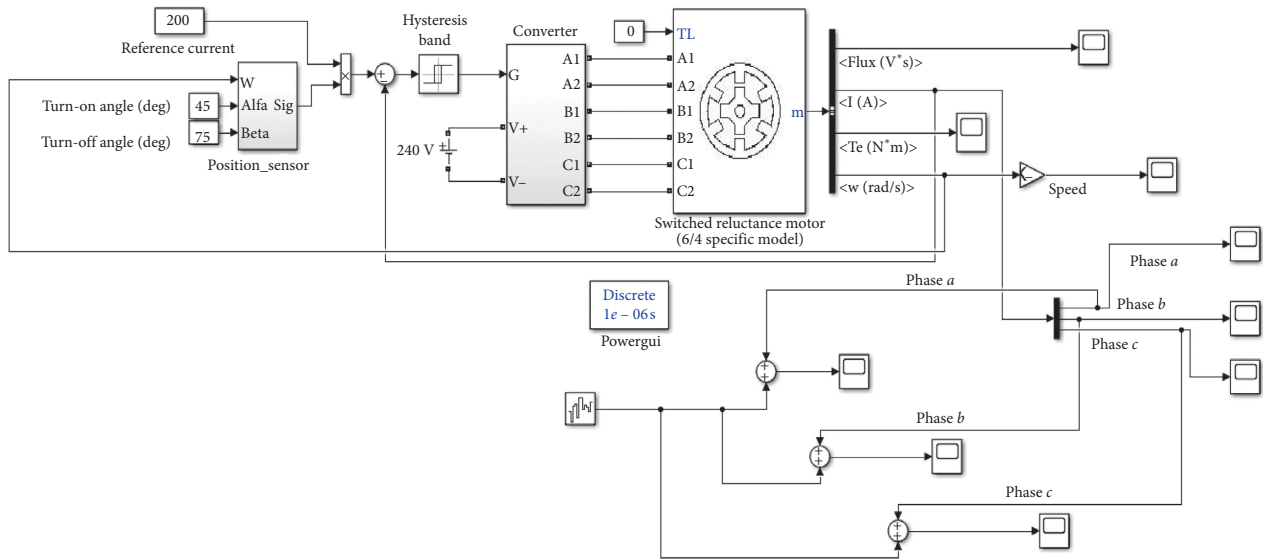
FIGURE 8: Simplified model of SRM by adding noise with the current.

Figure 9 shows the current of phase *a* in the case of the simplified model.

Figure 10 shows distortion in the phase *a* waveform after adding noise and depicts negative values. Values of phase *a* noticeable at different time instants: at $t = 0.1$ s, 85.8 A; at $t = 0.25$ s, −16 A; and at $t = 0.4$ s, 25 A in comparison with the simplified model phase *a* waveform displayed in Figure 9 which has value of 45.54 A at $t = 0.1$ s, −0.116 A at $t = 0.25$ s, and 3.22 A at $t = 0.4$ s.

Figure 11 shows ripples after adding noise with phase *b* simplified model waveform. The magnitude of phase *b* increases in the initial time interval then its magnitude decreases. In Figure 12, distinct values of currents at $t = 0.1$ s, 0.25 s, and 0.4 s are −0.121 A, 11.66 A, and 16.04 A whereas noise added waveform of phase *b* values at same time are 39.9 A, −4 A, and 37.8 A, respectively.

Figure 13 displays distortions due to addition of noise. Values of phase *c* current at $t = 0.1$ s, 0.25 s, and 0.4 s are 50 A, 30.5 A, and 21.7 A which means that noise added waveform of phase *c* does not display any negative values like noise added phase *a* and noise added phase *b* waveform. On the other side, values noticed in Figure 14 show the values of 9.99 A, 45.9 A, and −0.120 A.

Values of MSE are noticeable in the graphical results of simplified model waveforms of phase *a*, phase *b*, and phase *c*, and graphical results of noise added waveforms of phase *a*, phase *b*, and phase *c*, respectively, have same MSE values of 250.33% as shown in Table 2.

5.2. Results through NN Fitting Tool. Once the training is completed, the information provided by nntraintool is used to analyse the results.

Retraining several times for speed, phase *a*, phase *b*, and phase *c* data is conducted until the conditions of low MSE and higher regression (R) value of training, validation, and testing is achieved as depicted in Figures 15 and 16, respectively. Mean square error is the network performance function. It measures the performance of the network according to the mean of squared errors. MSE is the distance between the model's estimates of the test value. When MSE is small (closed to zero), it means that desired outputs for the training set have become very close to each other.

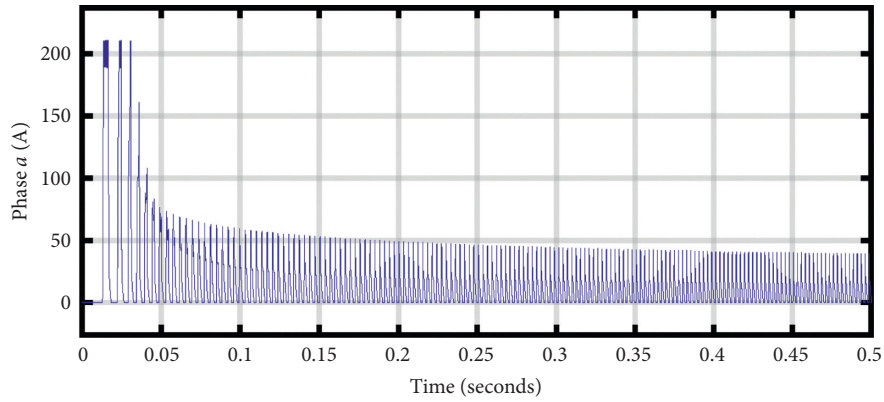The performance plot of speed data is described in Figure 17.
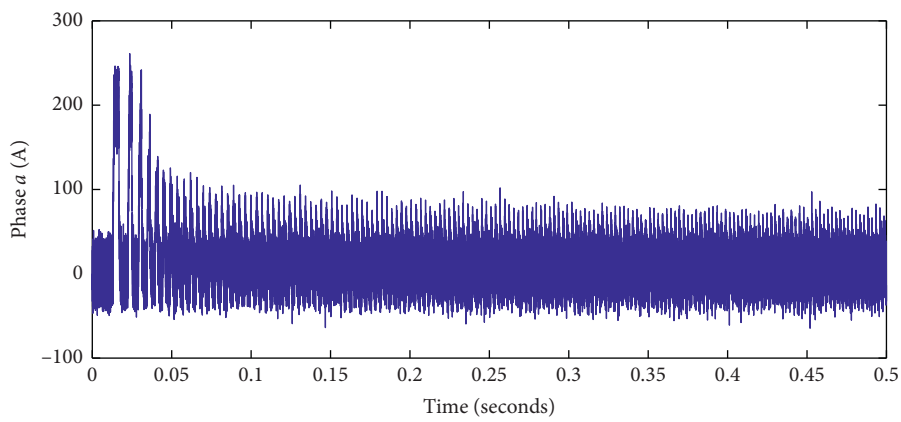
Figure 9: Phase *a* waveform of the simplified model.

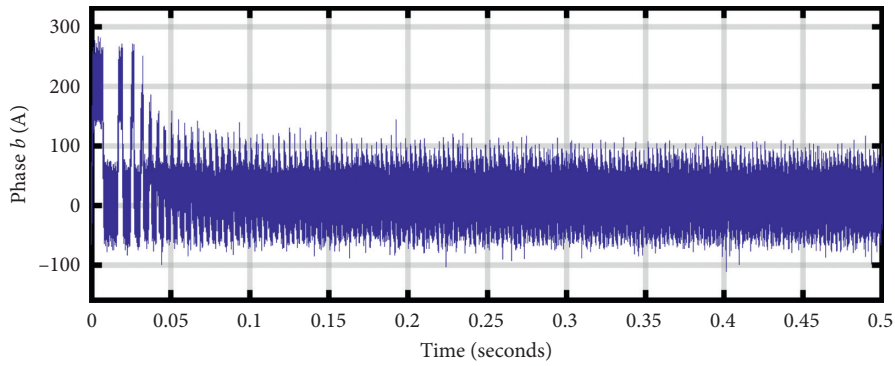Figure 10: Phase *a* noise waveform of the simplified model.

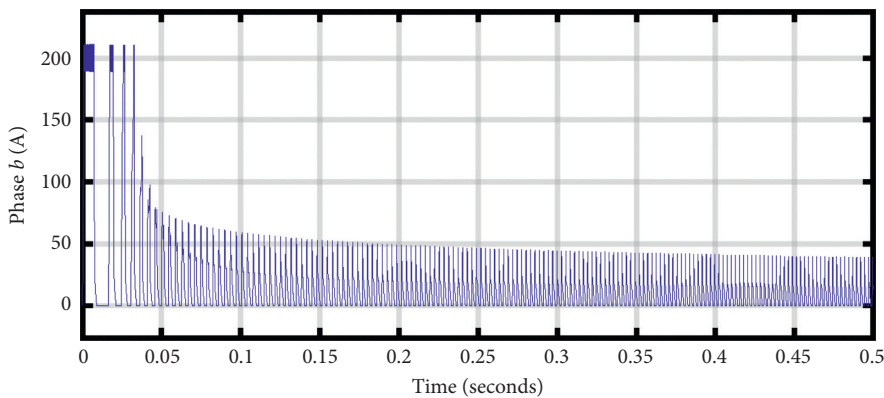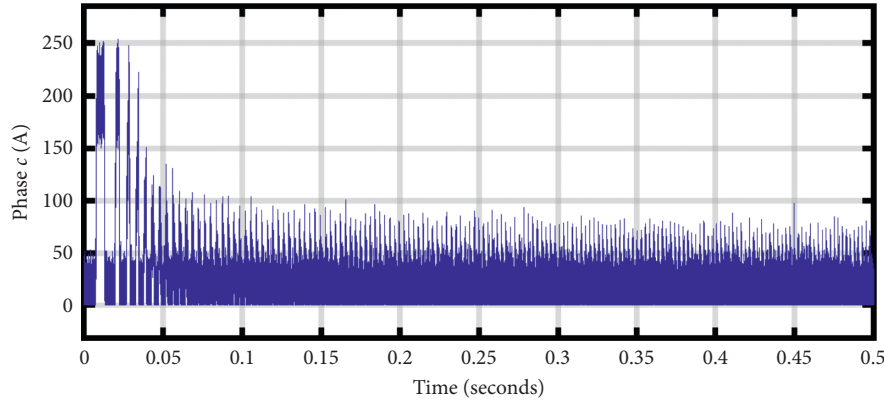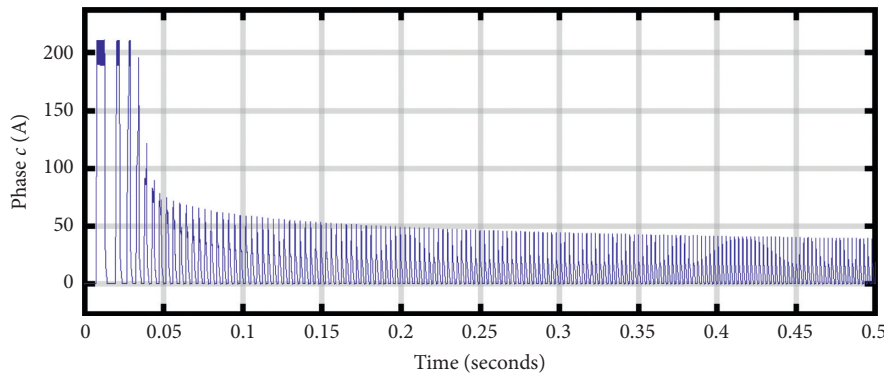Figure 11: Phase *b* noise waveform of a simplified model.

Figure 12: Phase *b* waveform of the simplified model.

FIGURE 13: Phase *c* noise waveform of a simplified model.

FIGURE 14: Phase *c* waveform of the simplified model.

TABLE 2: Values of the simplified model in different cases at $t = 0.1$, 0.25, and 0.4 seconds.

Sr. no.	Cases	MSE	Torque (Nm)		Speed (rad/s)	Phase *a* (A)	Phase *b* (A)	Phase *c* (A)
			Time (s)	—	—	—	—	—
1	Speed + noise and current simple	$2.275e + 07$	$T = 0.1$	117	4229	45.54	−0.121	9.996
			$T = 0.25$	52	5175	−0.116	11.66	45.9
			$T = 0.4$	17	5760	3.22	16.04	−0.120
2	Current noise and speed simple	250.338	$T = 0.1$	117	4189	85.5	39.9	50
			$T = 0.25$	52	5192	−16	−4	30.5
			$T = 0.4$	17	5735	25	37.8	21.7

Performance plot shows the graph of error versus epochs (one forward and backward pass) for training, validation, and test performances of training records. Mostly, after more epochs of training, the error reduces but might start to increase on the validation data set as the network starts overfitting the training data. In the default setup, after six consecutive increases in validation error, the training stops, and the best performance is taken from the epoch with the lowest validation error. It can be seen in Figure 17 that ANN fits the best value 0.47041 after 1 epoch (1 epoch = one forward and backward pass), the iteration at which validation reached a minimum. In Figure 18, phase *a* best validation performance is 0.50367 at epoch 21. Figure 19 exhibits the case of phase *b*. The best validation performance is 0.45799 at epoch 37.

Figure 20 shows the best-fitting value of 3.4557 at epoch 1 for phase *c* obtained by ANN. There are four lines: train, validation, test, and best. In fact, the best (dotted) line represents that other lines should lie on or near this (dotted) lines, and then it can be confirmed that training has been done successfully. If any of the three (training, validation, and testing) lines meet or pass near the best (dotted) line, it means convergence has been done, and if this is not the case, retrain the network. As seen in the performance plot of speed, phase *a*, phase *b*, and phase *c* data, MSE of ANN has decreased with the number of epochs. The ANN was well trained, so it presents low MSE at the end of the training phase.

Then, in order to check how well the input (speed and phase data with ripples) and output data (speed and phase data without ripples) are fitted in the network, the regression
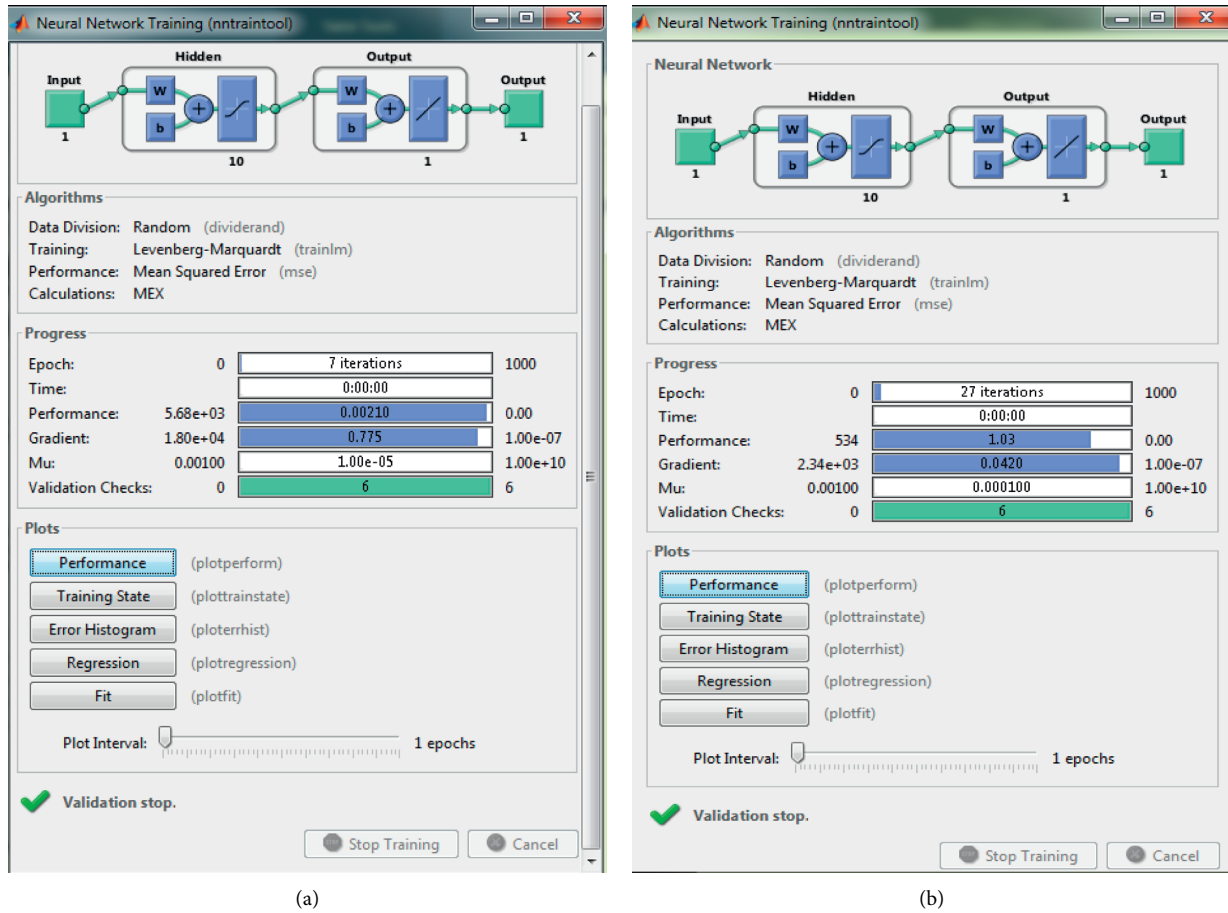
FIGURE 15: NNtraintool for (a) speed data and (b) phase *a* data.

plot is used. During training, 70% of data is used for training, 15% data is used for validation, and 15% data is used for testing the network. All these split up happen by default. The regression graph has four graphs for training, validation, testing, and combining all. The learning curve of neural network can be compared with the best training algorithm. Figure 21 shows the regression analysis of speed data between the response of the network and the desired objectives. The value of *R* shows the correlation coefficient between the outputs and the objectives. The correlation between the outputs and targets is very high due to all the *R* (regression) parameters are in proximity of 1. As seen clearly in the regression plot, speed data falls along a 45-degree line. It fulfils the condition of perfect fitted data. In a perfect fitted plot, the network outputs are equal to targets values. This plot has a perfect correlation in both cases, in the proximity of 1. All this is a measure of how well the output is adapted to the target.

Figure 22 illustrates the regression plot for phase *a* data and values of regression in case of training, validation, and testing.

Tables 3 and 4 show values of *R* and MSE of speed data and phase *a*, and *R* and MSE values are shown in Tables 5 and 6 in the case of different parameters of training, validation, and testing data. Overall, higher regression (*R*) value and low MSE value of phase *a* and speed data are achieved in the case of parameter combination of 70% training, 15% validation, and 15% testing.

Figure 23 shows the regression plot for phase *b*, and Figure 24 describes the regression graph for phase *c*.

Phase *b* values of *R* and MSE after randomly dividing the data at different combinations are explained in Tables 7 and 8, respectively, but better results are achieved at training = 70%, validation = 15%, and testing = 15%.

Tables 9 and 10 present phase *c* values achieved at different combination of data division.

5.3. Improved Model. Further artificial neural network is developed for the reduction of torque ripples and to accelerate the performance of the switched reluctance motor. ANN blocks of speed and phase *a*, phase *b*, and phase *c* that are the generated Simulink results of the trained neural network at the end of training are used for the reduction of ripples in the waveform of torque, speed, and current by regulating its speed and current. The model is simulated for the same time period *t* = 0.5 seconds and the same Simulink parameters described in Table 1. Four cases of the improved model are discussed here.

5.3.1. Improved Model Case 1. Figure 25 shows the ANN-based speed block in which speed of the motor is given as an input to ANN-based speed block and its output is connected to the reference speed of the position sensor and current is
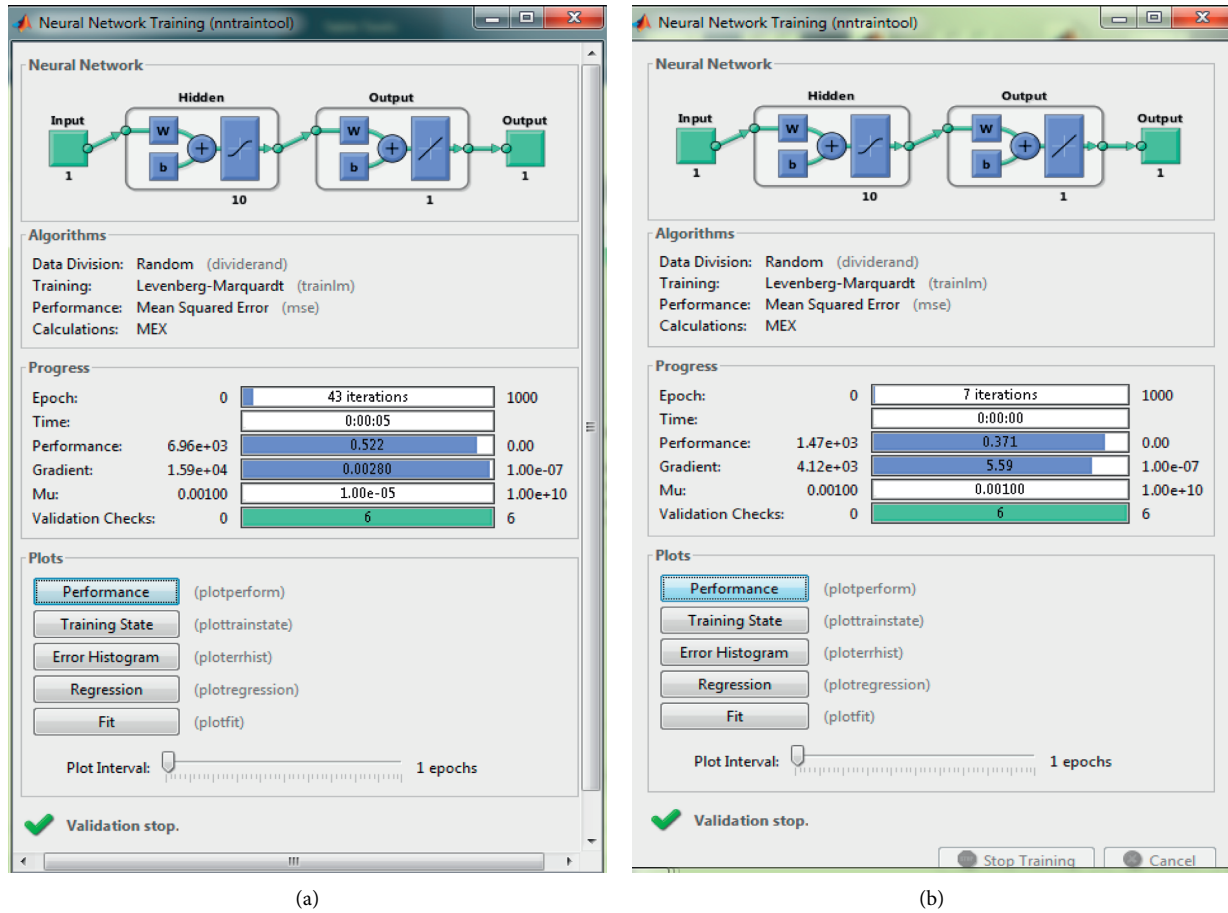
(a)   (b)

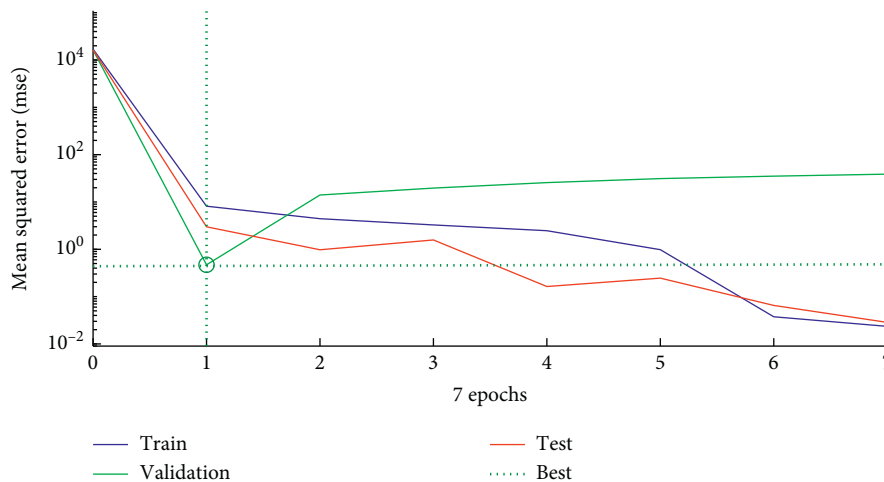Figure 16: NNtrain tool for (a) phase b data and (b) phase c data.

Figure 17: ANN performance plot for speed data.

simply untouched in the same manner as in the simplified model.

Figure 26 shows the waveform of simple speed data given as an input to the Simulink generated block of neural network trained data of speed.

Figure 27 shows the output waveform of speed achieved after employing the ANN speed block with untouched

current. Speed value is always absolute. Positive or negative value of speed does not matter. It just shows direction. Positive value shows increase in speed value, and negative value shows decrease in speed value. It shows fluctuation from higher to lower values [30, 31]. The mean value of speed (Figures 26 and 27) is positive. Comparison of graphical results of both input and output speed waveform at the time
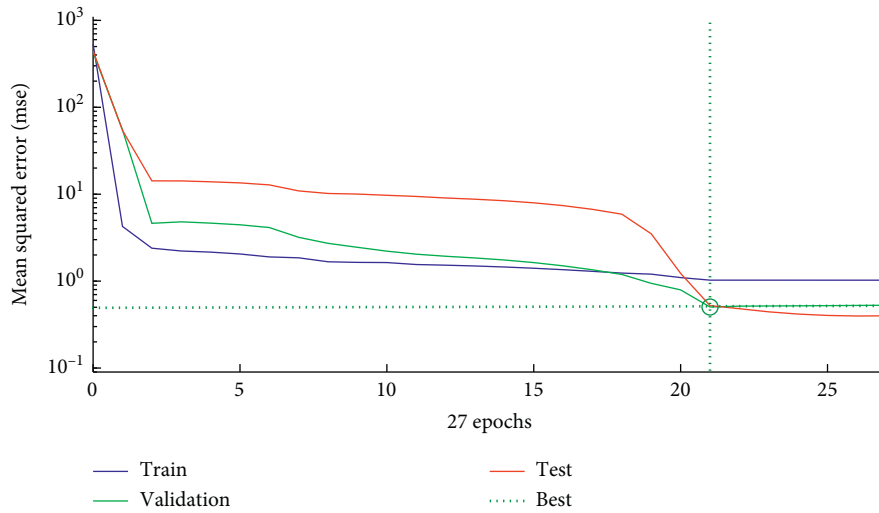
Figure 18: ANN performance plot for phase *a* data.
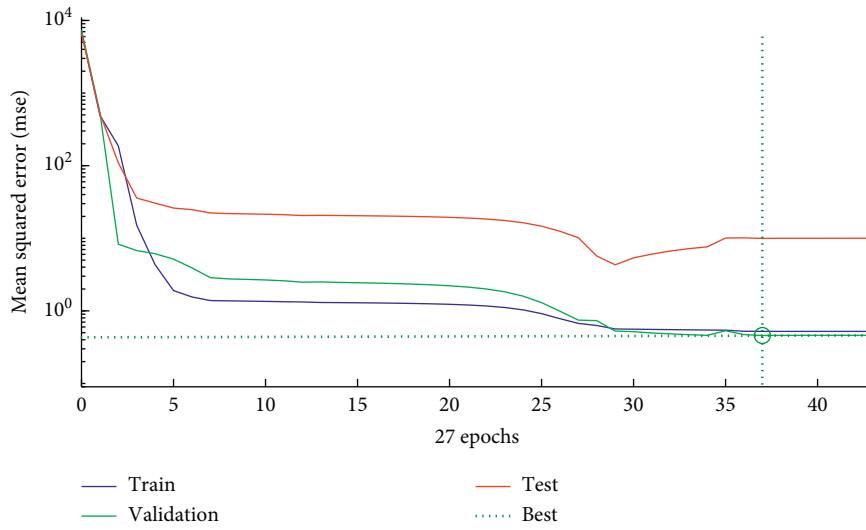
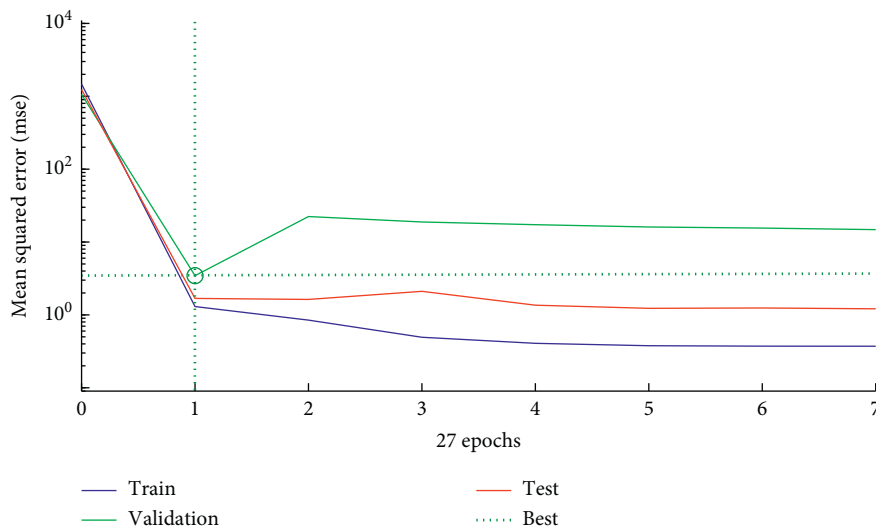Figure 19: ANN performance plot for phase *b* data.

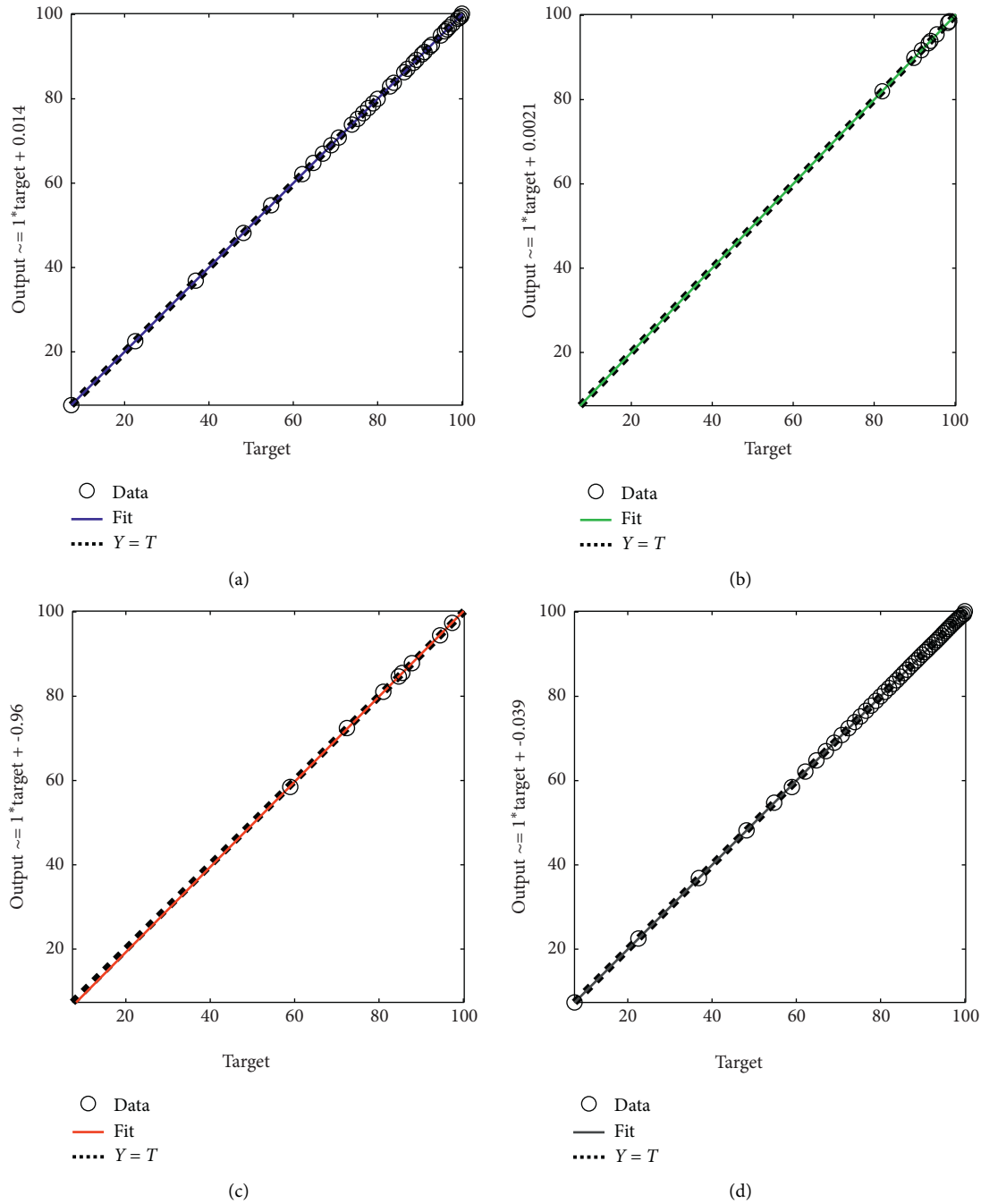Figure 20: ANN performance plot for phase *c* data.

(a)

(b)

(c)

(d)

FIGURE 21: Regression plot for speed data: (a) training: $R = 1$, (b) validation: $R = 0.99998$, (c) test: $R = 0.99995$, and (d) all: $R = 0.99999$.

$t = 0.1$ s, 0.25 s, and 0.4 s shows that the values of speed before ANN are 108 rad/s, 179 rad/s, and 78.8 rad/s, respectively, while these values of speed are increased to 1036 rad/s, 171 rad/s, and 752.9 rad/s in the case of output speed waveform after application of ANN. MSE between input and output speed waveform is $2.9618e + 04$%. Speed values at different steps are summarized in Table 11.

5.3.2. Improved Model Case 2.
In Figure 28, ANN blocks are added with each phase of the current in the simplified model

and checking its effect in the case of input and output waveforms of 3-phase current. 3-phase graphical results show the maximum fluctuation in current values during the starting time interval 0 s to 0.1 s, and that is why in Figures 29–34, for better understanding, current values during this time duration are exhibited.

R-value for phase a is 0.99092, and MSE noticeable in the graphical results of Figure 29 and improved waveform of phase a data given as input to the ANN block as displayed in Figure 30 is $1.8346e - 10$%. Graphical results of phase a improved waveform show that at $t = 0.1$ s, 0.25 s, and 0.4 s

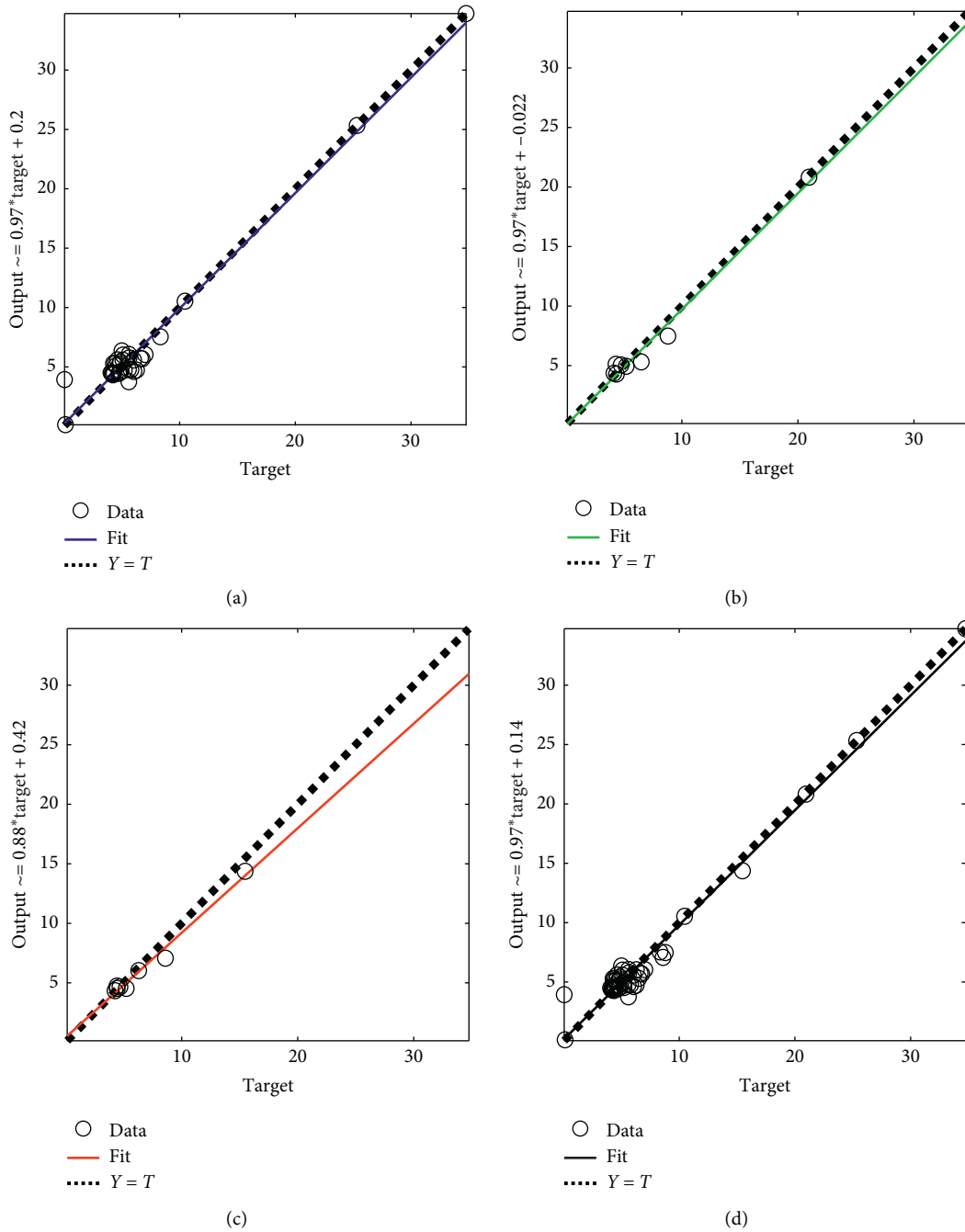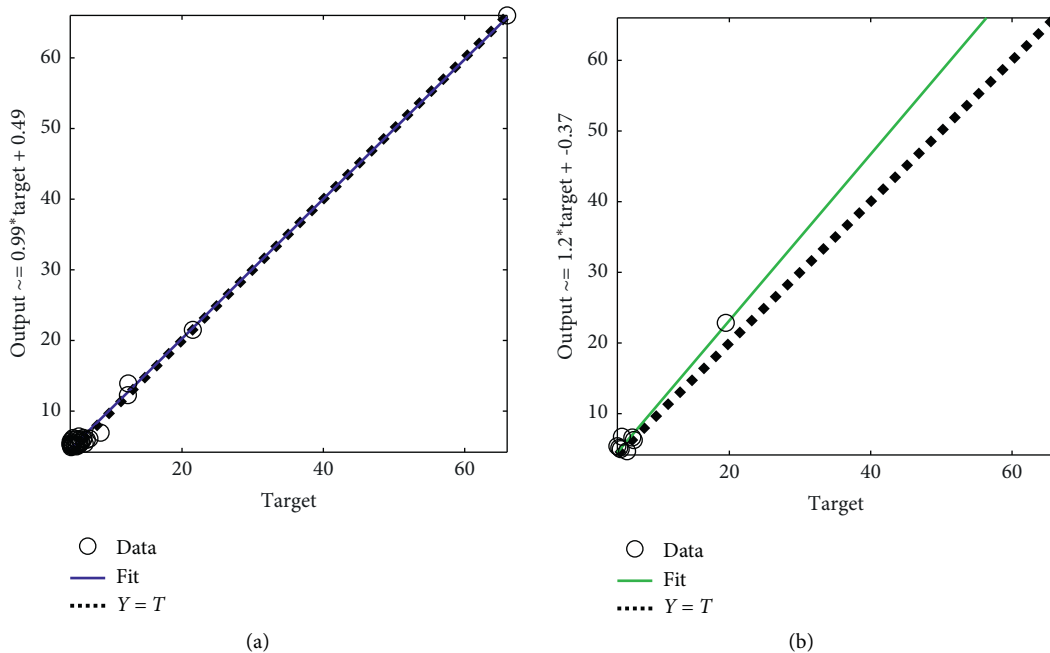Figure 22: Regression plot for phase *a* data: (a) training: $R = 0.98634$, (b) validation: $R = 0.99192$, (c) test: $R = 0.99104$, and (d) all: $R = 0.98674$.

Table 3: Regression values of speed data in different parameters.

Layers	Parameters	Training	Validation	Testing	All
10	60, 20, 20	0.9989	0.9989	0.9989	0.9997
10	70, 15, 15	0.9999	0.9997	0.9998	0.9998
10	80, 10, 10	0.9978	0.9980	0.9986	0.9978
10	90, 5, 5	0.9954	1.00	0.9865	0.9996
10	50, 5, 5	0.9987	0.9998	0.9945	0.9995

TABLE 4: MSE values of speed data in different parameters.

Layers	Parameters	Training	Validation	Testing
10	60, 20, 20	0.0231	0.0821	0.0923
10	70, 15, 15	0.0756	0.0337	0.0760
10	80, 10, 10	0.0980	0.0342	0.1428
10	90, 5, 5	0.8130	0.0542	0.0212
10	50, 5, 5	0.0560	0.2993	0.0392

TABLE 5: Regression values of phase a data in different parameters.

Layers	Parameters	Training	Validation	Testing	All
10	60, 20, 20	0.9855	0.9538	0.9873	0.9816
10	70, 15, 15	0.9863	0.9919	0.9910	0.9867
10	80, 10, 10	0.9910	0.9943	0.9677	0.9771
10	90, 5, 5	0.9776	0.9608	0.9909	0.9771
10	50, 5, 5	0.9787	0.9601	0.9924	0.9566

TABLE 6: MSE values of phase a data in different parameters.

Layers	Parameters	Training	Validation	Testing
10	60, 20, 20	0.9260	0.3111	4.4711
10	70, 15, 15	0.9056	0.7581	0.9957
10	80, 10, 10	0.5154	3.8314	6.9964
10	90, 5, 5	1.3398	0.7961	7.4414
10	50, 5, 5	1.2983	0.2398	10.903

(a)

(b)

FIGURE 23: Continued.

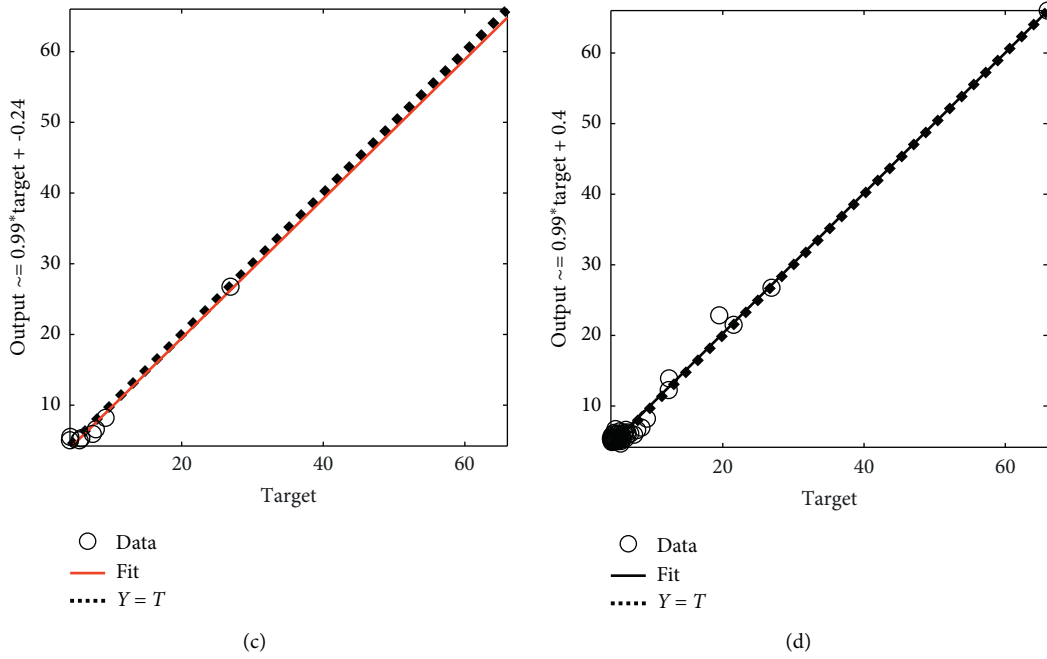Figure 23: Regression plot for phase *b* data: (a) training: $R = 0.99773$, (b) validation: $R = 0.9878$, (c) test: $R = 0.99142$, and (d) all: $R = 0.99518$.
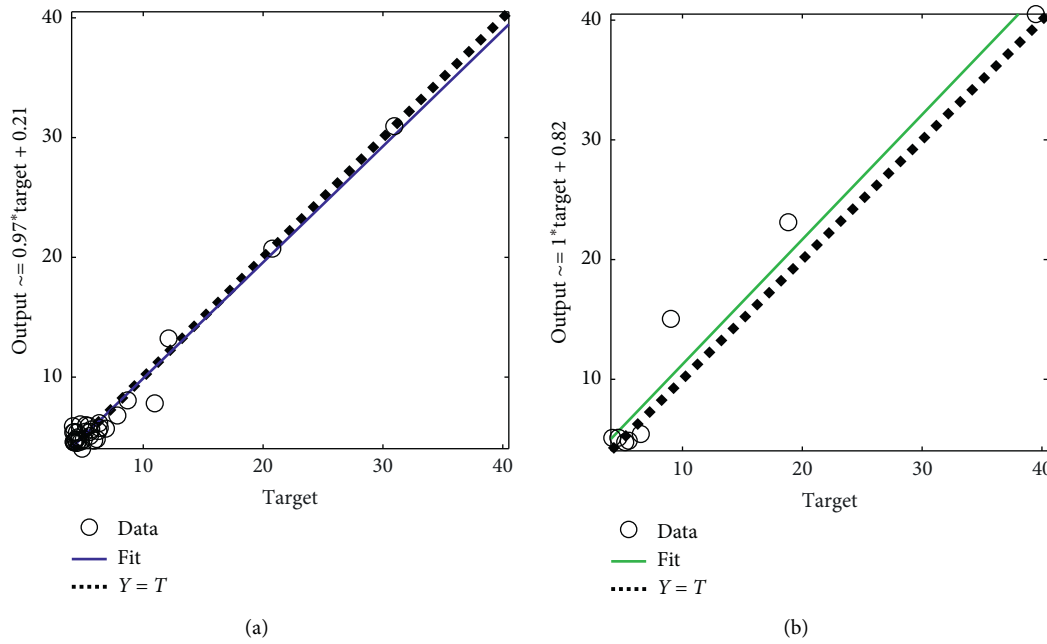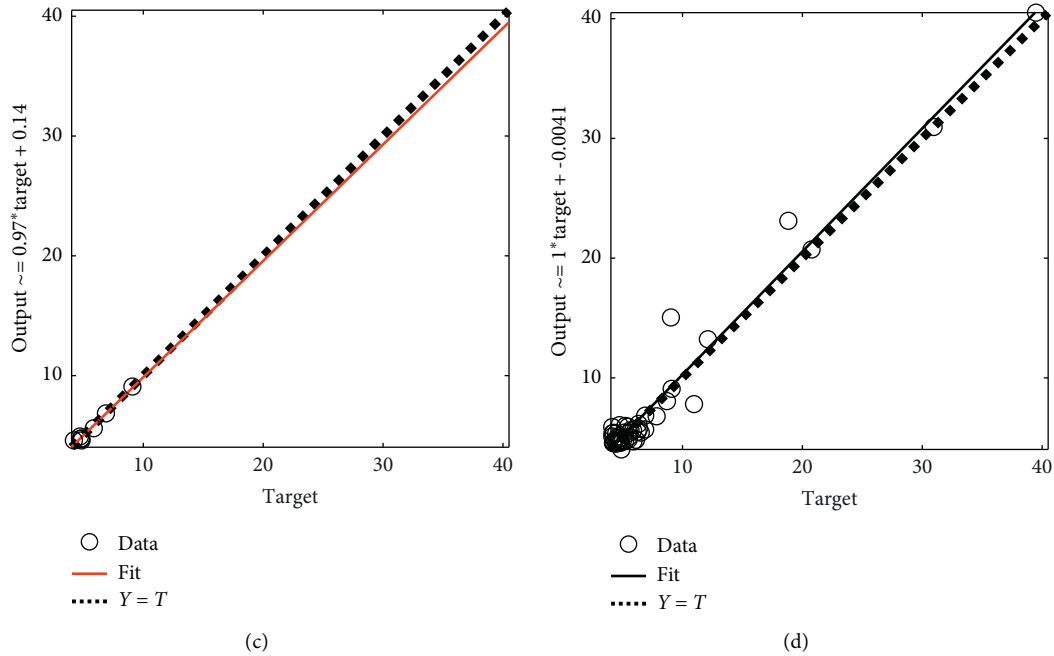

Figure 24: Continued.

(c)

(d)

FIGURE 24: Regression plot for phase c data: (a) training: $R = 0.984$, (b) validation: $R = 0.9817$, (c) test: $R = 0.99092$, and (d) all: $R = 0.98208$.

TABLE 7: Regression values of phase b data in different parameters.

Layers	Parameters	Training	Validation	Testing	All
10	60, 20, 20	0.9967	0.9920	0.9813	0.994
10	70, 15, 15	0.9977	0.9898	0.9914	0.9951
10	80, 10, 10	0.9881	0.9561	0.999	0.9887
10	90, 5, 5	0.9974	0.9312	0.9759	0.9962
10	50, 5, 5	0.9900	0.9885	0.9935	0.9725

TABLE 8: MSE values of phase b data in different parameters.

Layers	Parameters	Training	Validation	Testing
10	60, 20, 20	0.7846	2.60	0.759
10	70, 15, 15	0.5685	0.998	0.967
10	80, 10, 10	2.283	1.729	0.8762
10	90, 5, 5	0.5076	2.143	1.575
10	50, 5, 5	30.48	52.61	41.49

TABLE 9: Regression values of phase c data in different parameters.

Layers	Parameters	Training	Validation	Testing	All
10	60, 20, 20	0.9804	0.9958	0.9866	0.9457
10	70, 15, 15	0.984	0.9817	0.9909	0.9820
10	80, 10, 10	0.9953	0.9752	0.9599	0.998
10	90, 5, 5	0.9652	0.9812	0.943	0.9462
10	50, 5, 5	0.9412	0.972	0.9935	0.9688

TABLE 10: MSE values of phase c data in different parameters.

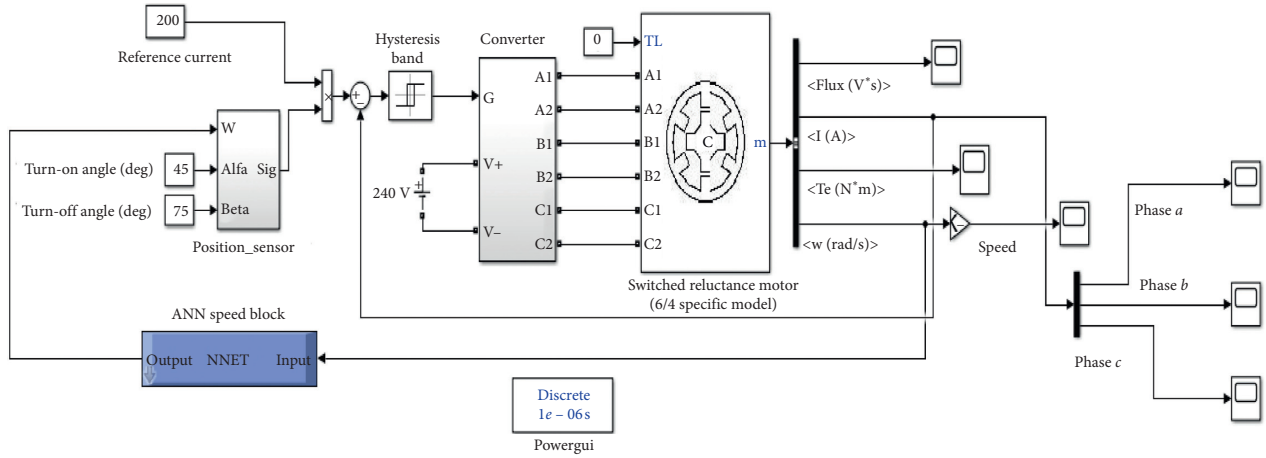| Layers	Parameters	Training	Validation	Testing
10	60, 20, 20	0.613	47.18	0.310
10	70, 15, 15	0.8493	0.893	0.0420
10	80, 10, 10	0.5192	0.130	2.47
10	90, 5, 5	0.580	5.60	0.607
10	50, 5, 5	0.474	0.028	23.94

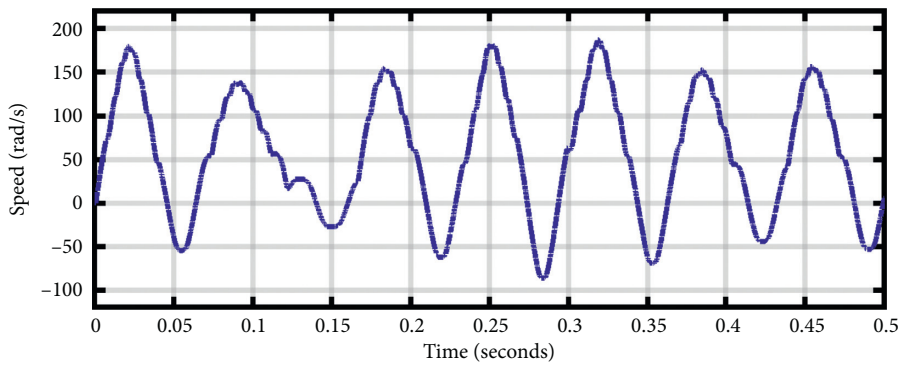Figure 25: Improved model of ANN-based speed and current simple.

Figure 26: Waveform of speed data as input to the ANN-based speed block.
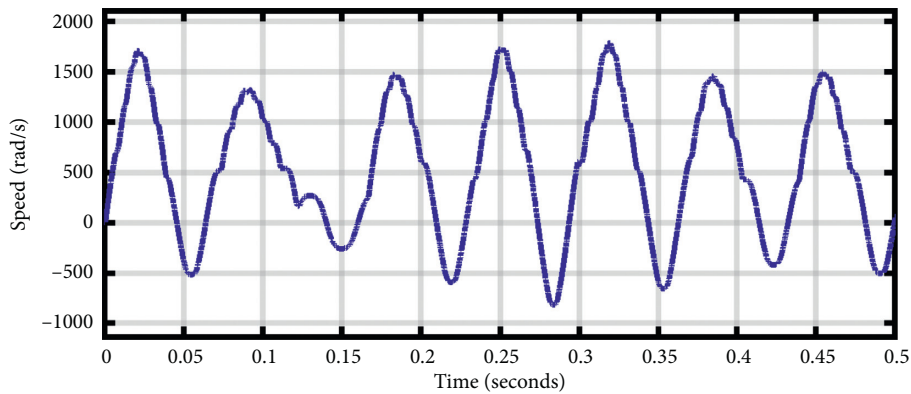

Figure 27: Desired output waveform of ANN-based speed and current simple.

Table 11: Improved model case 1 speed values at different steps.

Time (s)	—	Input speed before ANN (rad/s)	Output speed before ANN (rad/s)
0.1	—	108	1036
0.25	—	179	1715
0.4	—	78	752.9
—	Mean	5.752	54.92
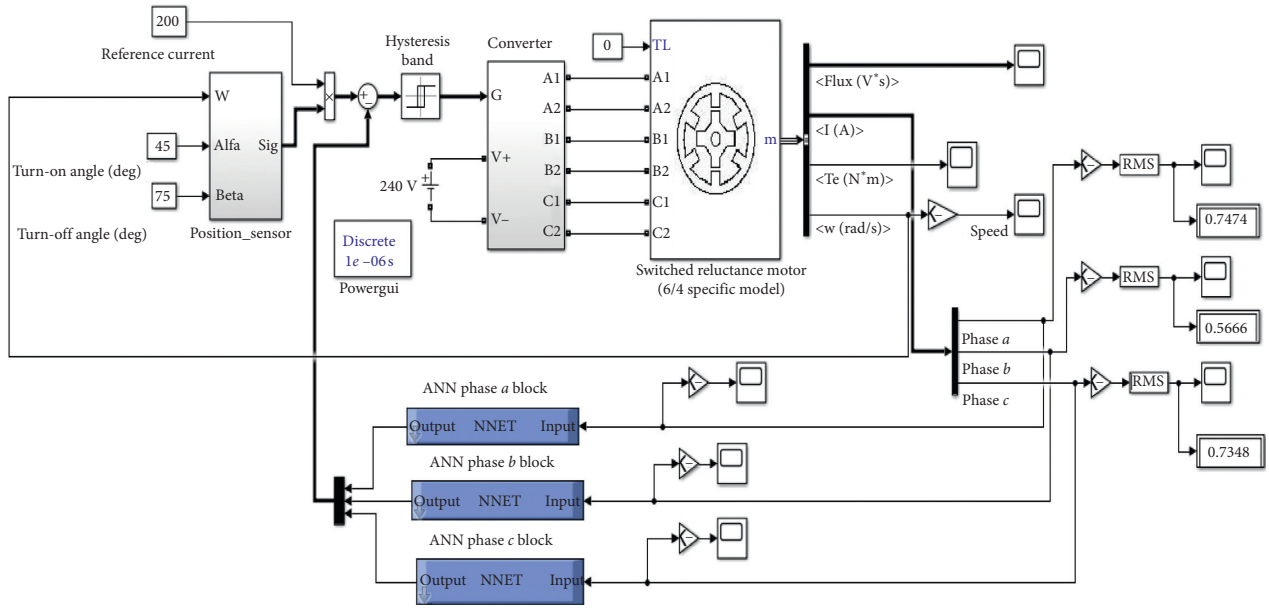
Figure 28: Improved model of ANN-based current and speed simple.
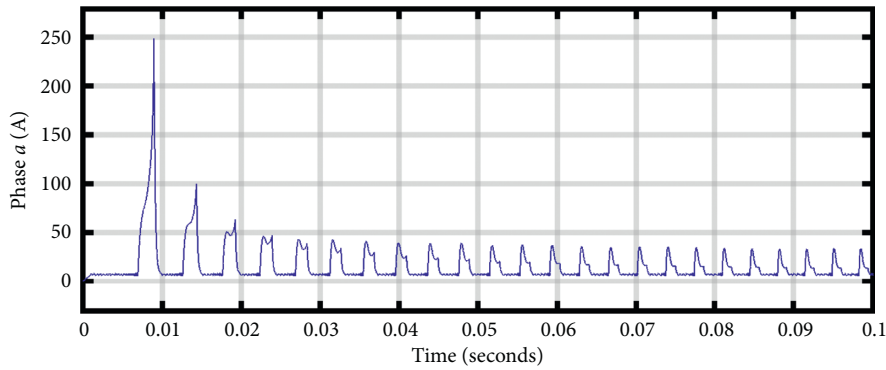
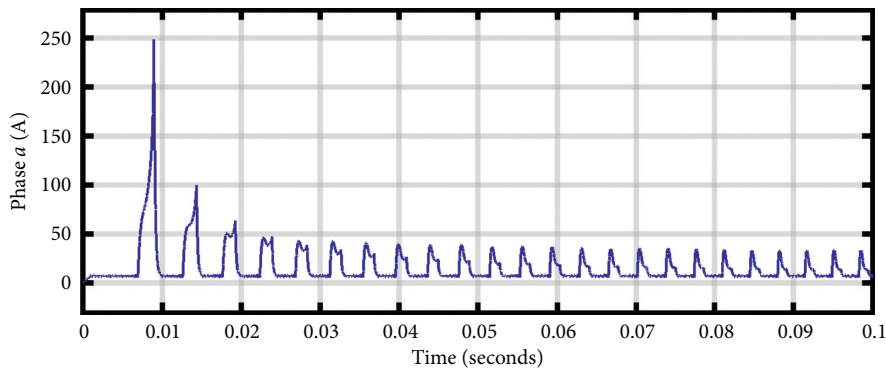Figure 29: Improved model phase *a* waveform.

Figure 30: Phase *a* input waveform.

values are 1.388 A, 7.78 A, and 11 A as compared with the values noticeable in the graphical result of phase *a* which has a current value of 6.95 A, 7.85 A, and 12.8 A at the same time points.

In Figure 31, *R*-value for phase *b* data is 0.99518, and MSE noticeable in the graphical results of phase *b* of the improved model and waveform of phase *a* data given as input to the ANN block as depicted in Figure 32 is
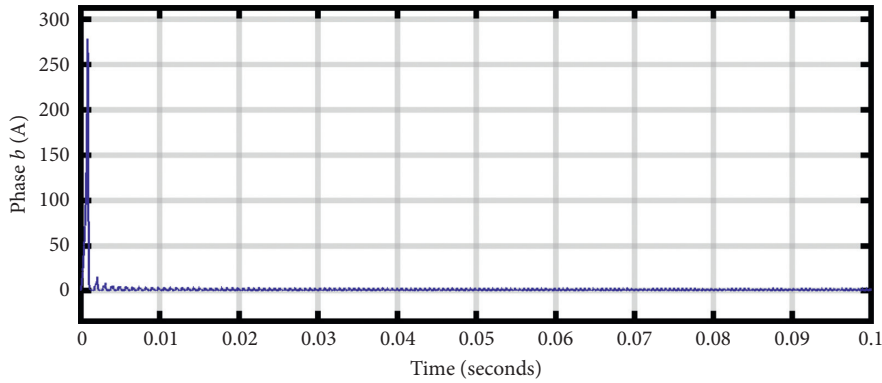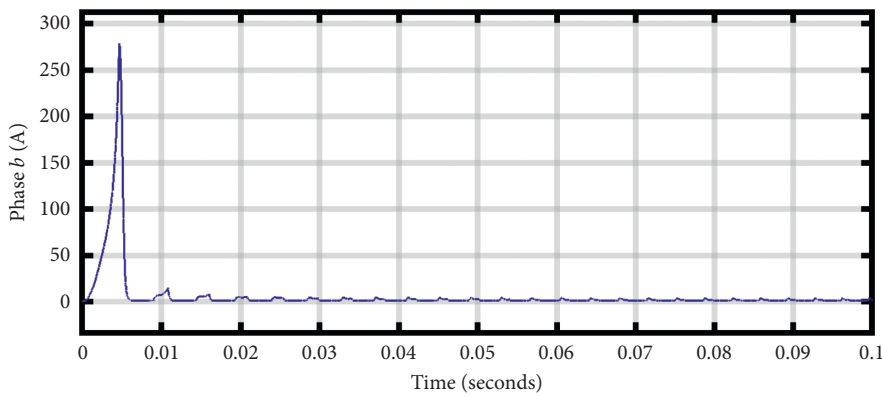
FIGURE 31: Improved model phase *b* waveform.
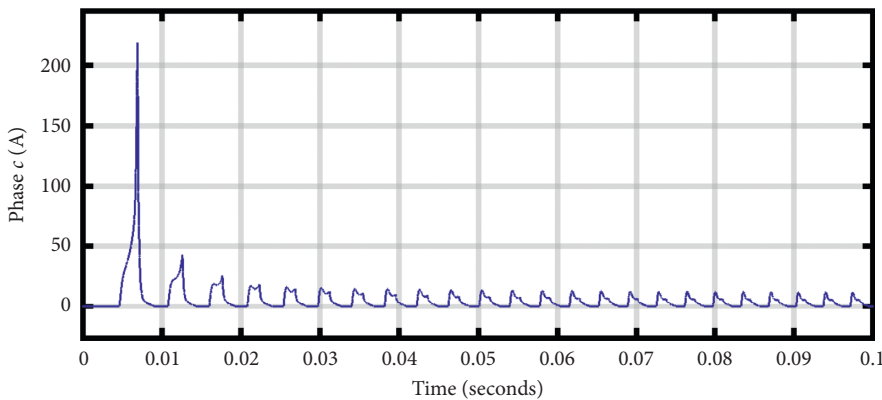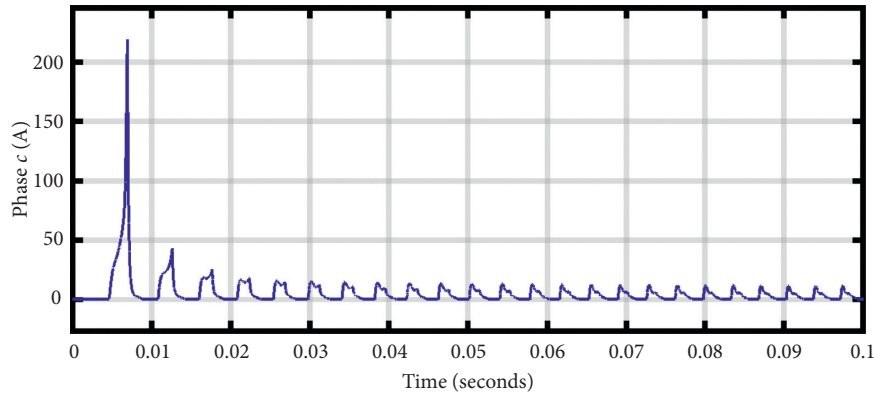

FIGURE 32: Phase *b* input waveform.

FIGURE 33: Improved model phase *c* waveform.

1.3625e − 11%. At time instants 0.1 s, 0.25 s, and 0.4 s, values in the case of input phase *b* waveform are 1.489 A, 0.925 A, and 0.455 A whereas values at the same points in the case of an improved waveform of phase *b* are 1.32 A, 0.923 A, and 0.439 A.

R-value for phase *c* is 0.9820, and value of MSE noticeable in the graphical results of phase *c* of the improved model as exhibited in Figure 33 and waveform of phase *c* data given as input to the ANN block as shown in Figure 34

is 5.4102e − 04%. The improved waveform of phase *c* has values of 0.0121 A, 0.025 A, and 3.23 A whereas values in the case of input waveform of phase *c* are −0.024 A, −0.0239 A, and 3.28 A at time 0.1 s, 0.25 s, and 0.4 s. Overall, the results of 3-phase current at different steps with change in the reference current are illustrated in Tables 12–14.

A band-limited white noise Simulink block is used in improved model case 3 and case 4, and the noise power value is 0.0005 W·s.

FIGURE 34: Phase c input waveform.

TABLE 12: Improved model case 2 phase a values at different steps.

Time (s)	Input phase a	—	Improved model phase a
0.1	6.95	—	1.388
0.25	7.85	—	7.78
0.4	12.8	–	11
—	—	RMS	0.7

TABLE 13: Improved model case 2 phase b values at different steps.

Time (s)	Input phase b	—	Improved model phase b
0.1	1.489	—	1.326
0.25	0.925	—	0.923
0.4	0.455	—	0.439
—	—	RMS	0.5

TABLE 14: Improved model case 2 phase c values at different steps.

Time (s)	Input phase c	—	Improved model phase c
0.1	−0.024	—	0.0121
0.25	−0.0239	—	0.025
0.4	3.28	—	3.23
—	—	RMS	0.7

5.3.3. Improved Model Case 3.

Figure 35 shows a scenario in which speed with noise data is given at the input and ANN-based speed block output is connected to the reference speed of the position sensor.

Figure 36 shows an improved output waveform of speed with the reduction of ripples and an increase in the speed in the same manner as compared with Figure 37 the input of speed with noise given as an input to the ANN speed block. The regression value of speed data is 0.99985 after 70% training, 15% validation, and 15% testing of speed data. In Table 15, comparison shows that values of speed are increased, and MSE noticeable in graphical results of speed graphs is $7.1480e + 05\%$.

5.3.4. Improved Model Case 4.

In Figure 38, ANN blocks are added individually with 3-phase current. In this case of improved model, time duration for graphical results of 3-phase current displayed in Figures 39–44 is from 0 s to 0.1 s but the simulation time of this model is 0.5 s.

Figure 40 shows phase a graphical result that has values of 11.5 A, 9.4 A, and 2.63 A at time intervals 0.1 s, 0.25 s, and 0.4 s. These values are reduced to 8.04 A, 5.91 A, and 0.039 A in the graphical result of phase a as displayed in Figure 39 after simulation of the model.

A comparison of graphical results of Figures 41 and 42 notifies that values are reduced in an improved graph of phase b rather in the case of phase b with noise given as an input. At $t = 0.1$ s, values are improved from 0.235 A to 0.0026 A; at $t = 0.25$ s, values are improved from 0.23 A to 0.205 A; and at $t = 0.4$ s, values are improved from 0.325 A to 0.145 A, respectively.

Figure 44 shows the input waveform of phase c with noise. Values observed in the graphical results at the same time $t = 0.1$ s, 0.25 s, and 0.4 s are 2 A, 1.4 A, and 2.71 A. Figure 43 shows reduced values of 0.1030 A, 0.3427 A, and
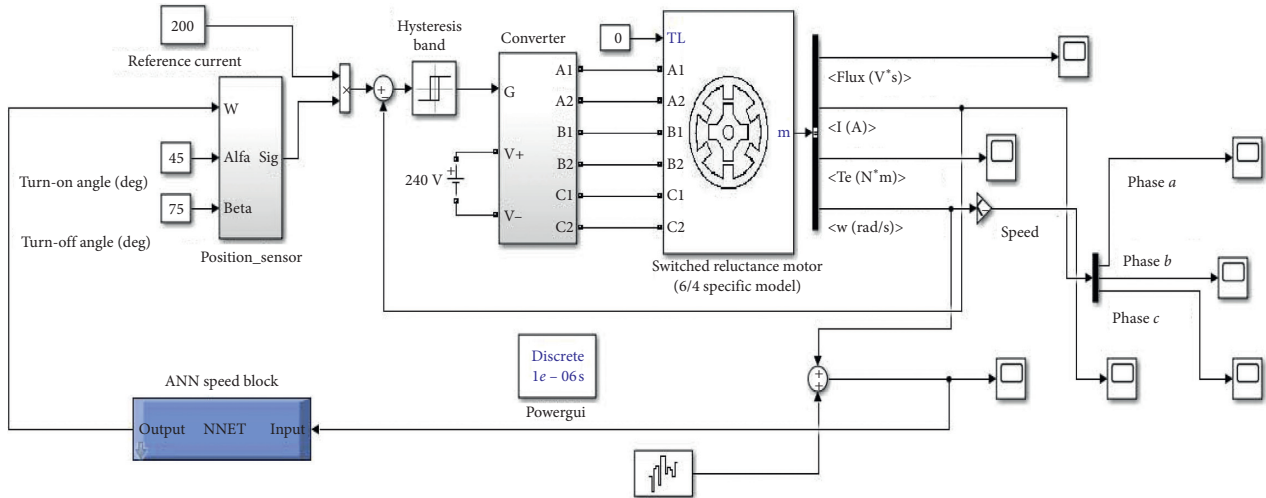
Figure 35: Improved model of ANN-based speed with noise and current simple.
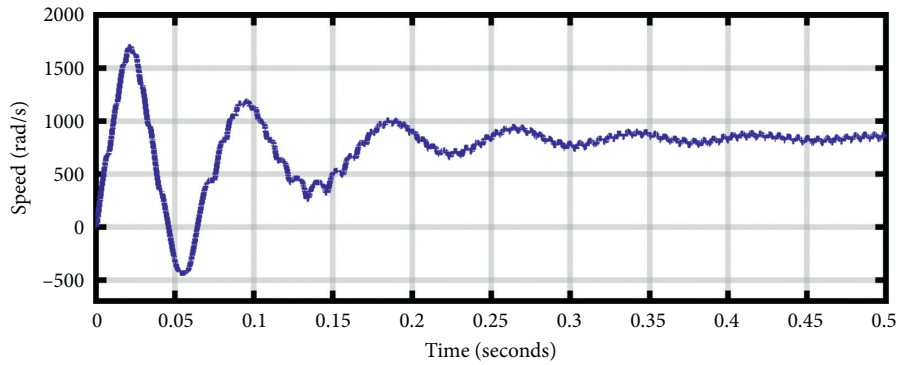

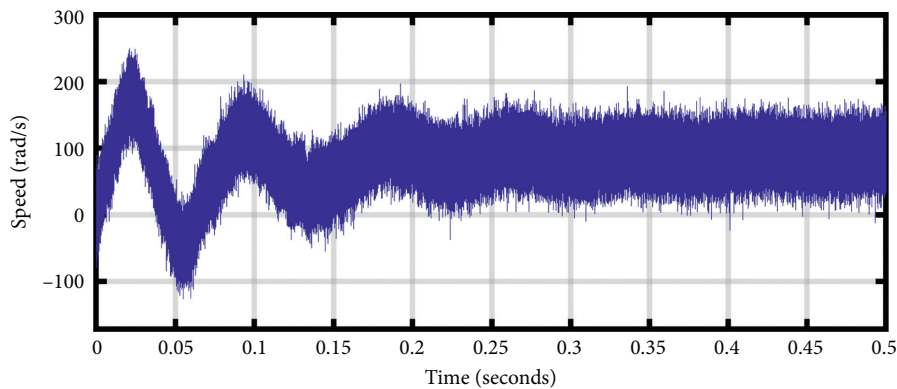Figure 36: ANN-based improved model speed.

Figure 37: Speed with noise as an input to the ANN block.

Table 15: Improved model case 3 speed values at different steps.

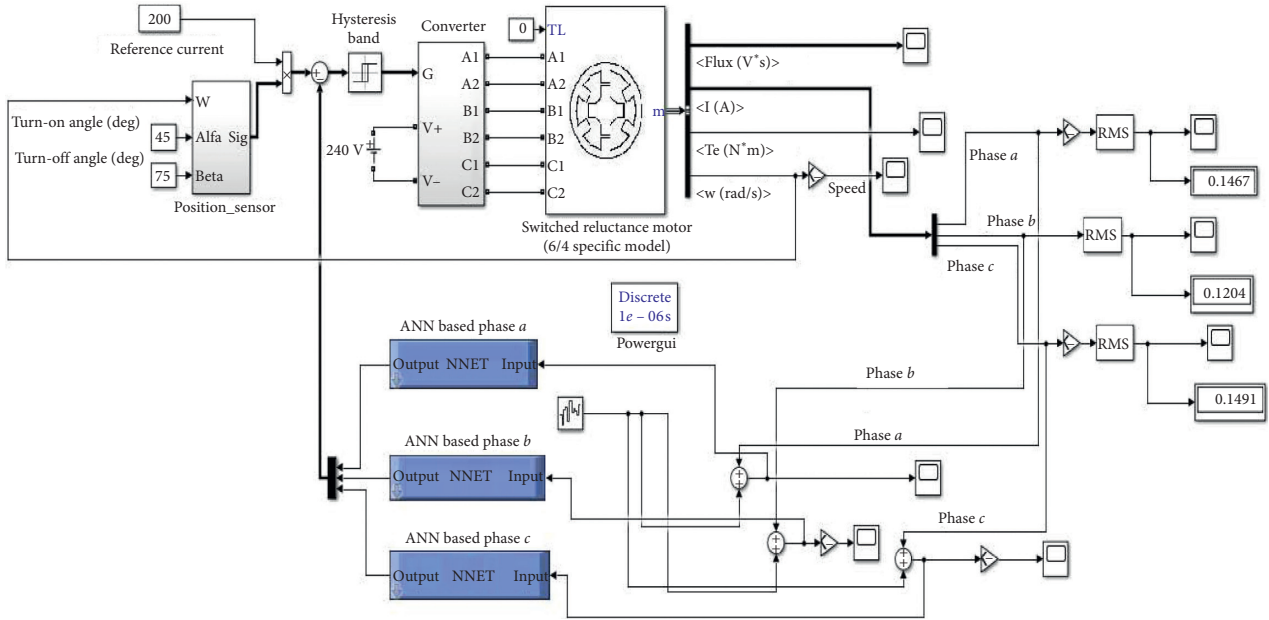| Time (s)	—	Speed with noise as input (rad/s)	Output speed before ANN (rad/s)
0.1	—	158	1122
0.25	—	72	821.3
0.4	—	106.5	811.2
—	Mean	98.24	848.5

Figure 38: Improved model of ANN-based current with noise and speed simple.
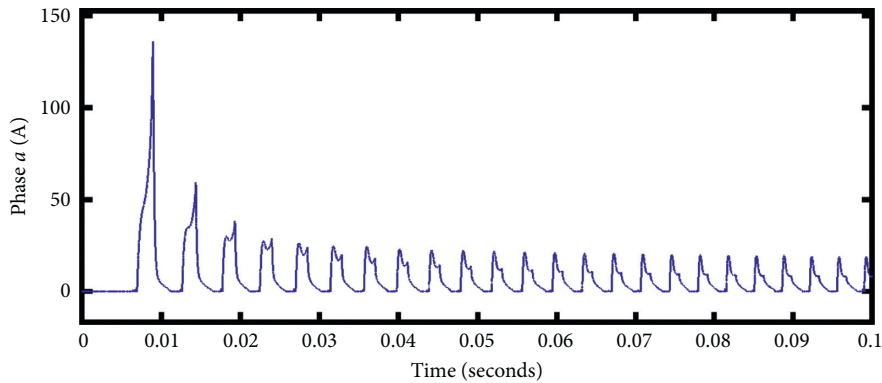

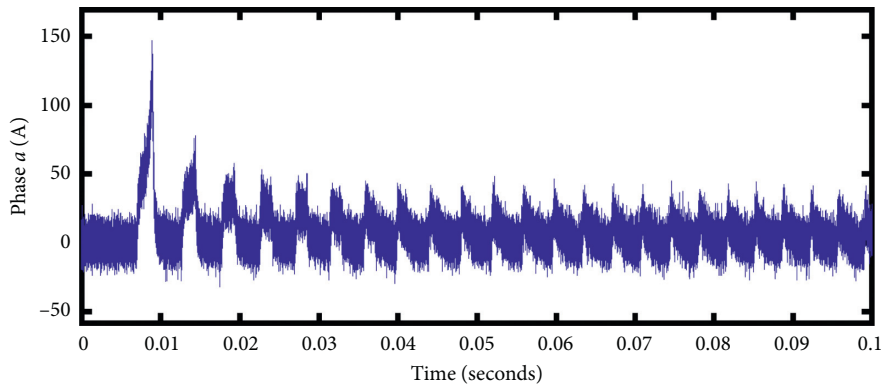Figure 39: Improved model phase *a* output waveform.

Figure 40: Phase *a* with noise as an input.

1.921 A at the same time points. Calculated RMS value of phase *a*, phase *b*, and phase *c* are 0.4 A, 0.1 A, and 0.1 A, respectively. Results of this case at different steps with change in the reference current are shown in Tables 16–18.

In case 4 of the improved model, the MSE value noticeable in the comparison with graphical results of 3-phase current (phase *a*, phase *b*, and phase *c*) after ANN and phase *a* (noise), phase *b* (noise), and phase *c* (noise) applied as an
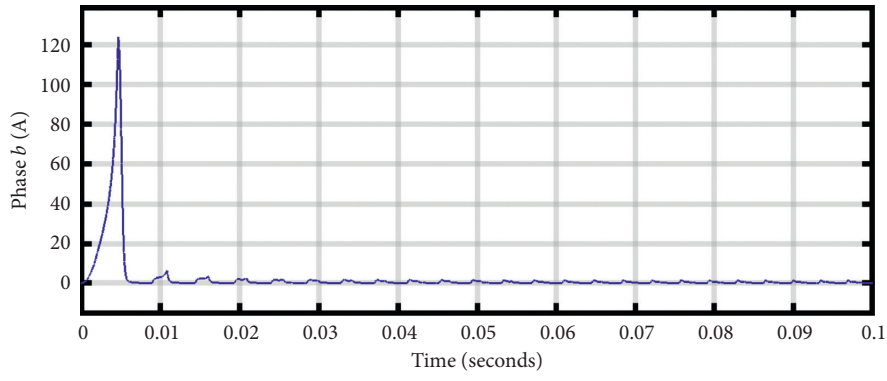
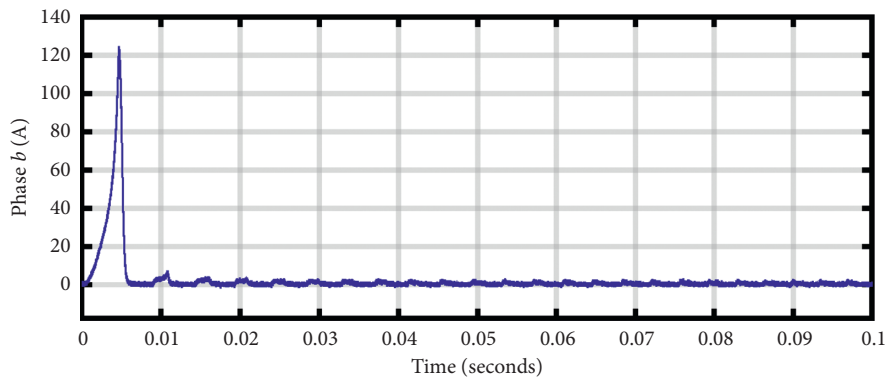FIGURE 41: Improved model phase *b* output waveform.

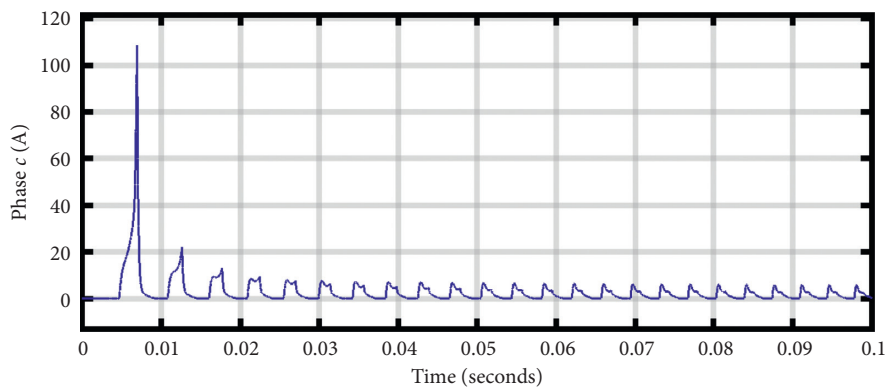FIGURE 42: Phase *b* with noise as an input.

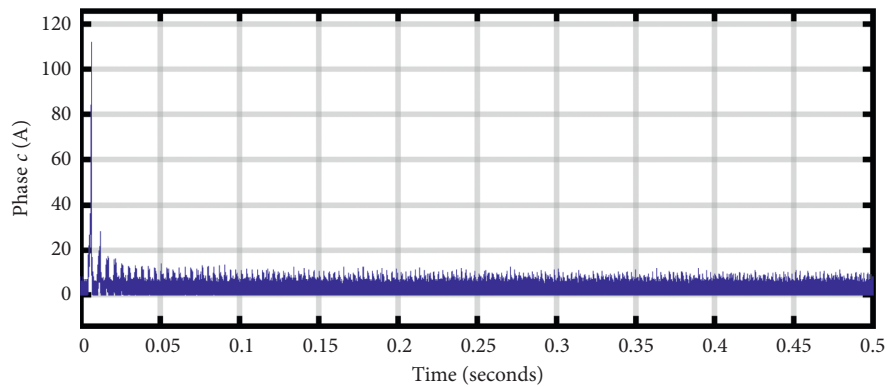FIGURE 43: Improved model phase *c* output waveform.

FIGURE 44: Phase *c* with noise as an input.

TABLE 16: Improved model case 4 phase *a* values at different steps.

Time (s)	Phase *a* with noise as input	—	Improved model phase *a*
0.1	11.5	—	8.04
0.25	9.4	—	5.91
0.4	2.63	—	0.039
—	—	RMS	0.1

TABLE 17: Improved model case 4 phase *b* values at different steps.

Time (s)	Phase *b* with noise as input	—	Improved model phase *b*
0.1	0.235	—	0.0026
0.25	0.238	—	0.205
0.4	0.325	——	0.145
—	–	RMS	0.1

TABLE 18: Improved model case 4 phase *c* values at different steps.

Time (s)	Phase *c* with noise as input	—	Improved model phase *c*
0.1	2	—	0.1030
0.25	1.4	—	0.342
0.4	2.71	—	1.921
—	—	RMS	0.1

TABLE 19: Values of the improved model in different cases at $t = 0.1$, 0.25, and 0.4 seconds.

Sr. no.	Cases	MSE	Torque (Nm)		Speed (rad/s)	Phase *a* (A)	Phase *b* (A)	Phase *c* (A)
			T (s)	—	—	—	—	—
1	Speed + ANN and current	2.9618*e* + 04	*T* = 0.1	−380	1036	−0.1197	147.16	2.0127
			T = 0.25	−0.63	1715	194	−0.119	−0.120
			T = 0.4	−412	752.9	210	−0.1198	−0.120
2	Current + ANN and speed	Phase *a*, 5.4102*e* − 04	*T* = 0.1	55.7	7386	1.388	2.99	0.0121
		Phase *b*, 1.3625*e* − 11	*T* = 0.25	51.3	8803	7.78	0.923	0.025
		Phase *c*, 1.8346*e* − 10	*T* = 0.4	48.4	9548	11	0.455	3.236
3	Speed + noise + ANN and current simple	7.1480*e* + 03	*T* = 0.1	1.44	1122.2	−0.120	200.1	−0.119
			T = 0.25	−0.67	821.3	−0.12	18.57	15.26
			T = 0.4	2.53	811.2	17	−0.119	16.39
4	Current + noise + ANN and speed simple	Phase *a*, 220.81	*T* = 0.1	53.2	8784	8.04	0.0026	0.103
		Phase *b*, 220.81	*T* = 0.25	52	10532	5.91	0.205	0.342
		Phase *c*, 220.81	*T* = 0.4	52.8	11588	0.039	0.145	1.921

input to ANN blocks of phase *a*, phase *b*, and phase *c*, respectively, is 249.81%, and overall *R*-value is 0.98631 for phase *a*, 0.99518 for phase *b*, and 0.99092 for phase *c* after training data of phase *a*, phase *b*, and phase *c* according to parameters of 70% training, 15% validation, and 15% testing.

Table 19 shows values of the improved model in different cases.

5.4. *ANN-Based SRM Improved Model.* In Figure 45, the ANN-based current block is added individually with each phase of three phases of current (noise) and the ANN-based speed block is added with speed (noise).

5.4.1. *Analysis of the ANN-Based SRM Model.* SRM draws high current in the starting interval proportional to torque. Motor starting and running characteristics during the time interval of 0.5 seconds are shown in Figures 46–50.

In SRM, speed and torque have an inverse relationship, graphical results prove that as there is a reduction in torque ripples, and speed increases, and this leads to the better performance of SRM. Stator 3-phase current (phase *a*, phase *b*, and phase *c*) characteristics are exhibited by the graphical results of the figures. In SRM, the current is directly proportional to torque while in conventional machines torque is directly proportional to the square of the current.
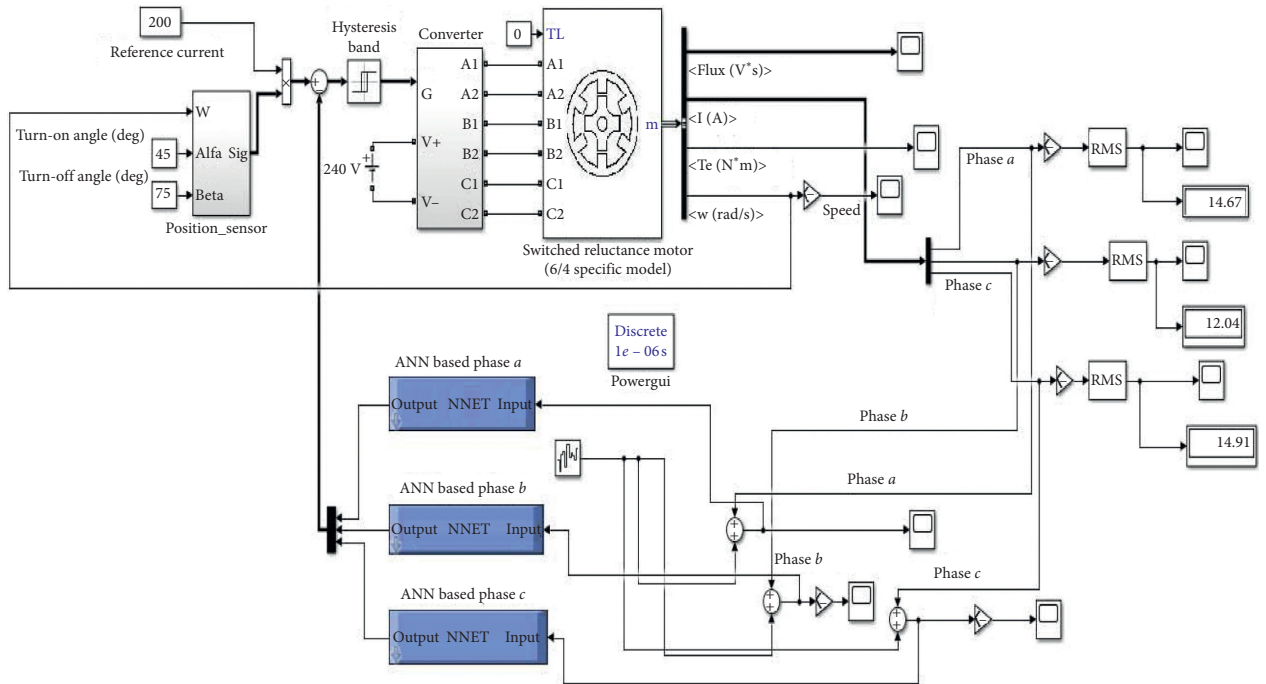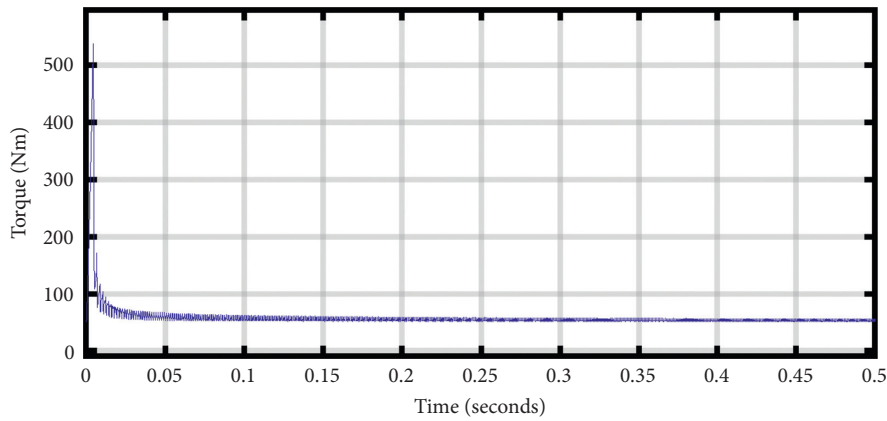
FIGURE 45: ANN-based SRM.

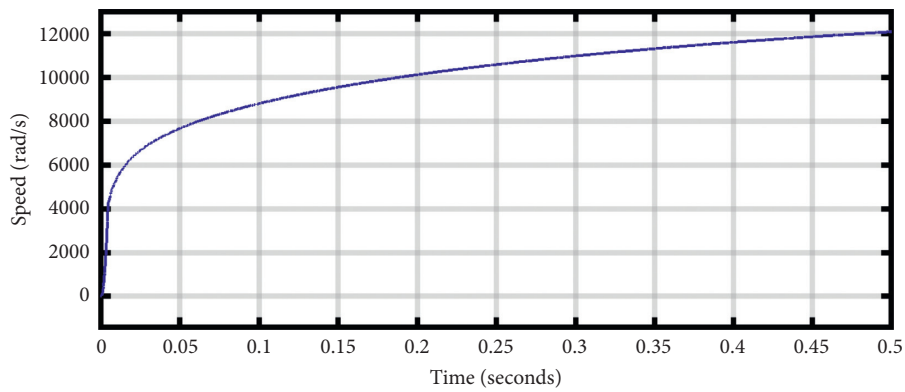FIGURE 46: The characteristics of torque in ANN-based SRM.

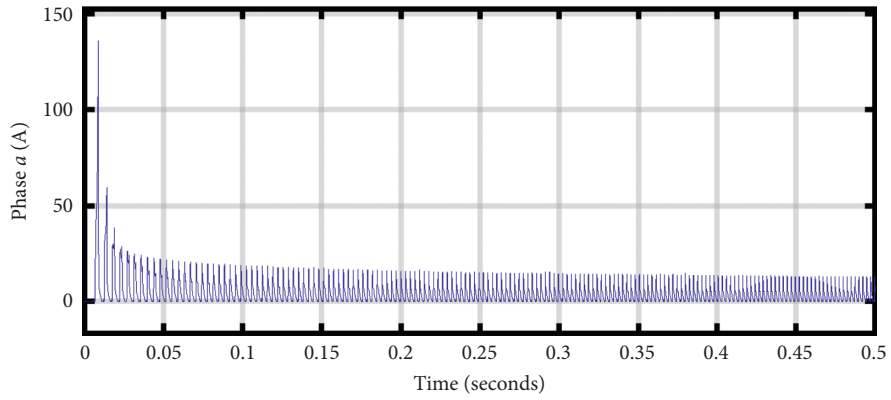FIGURE 47: Characteristics of building up of speed in SRM.

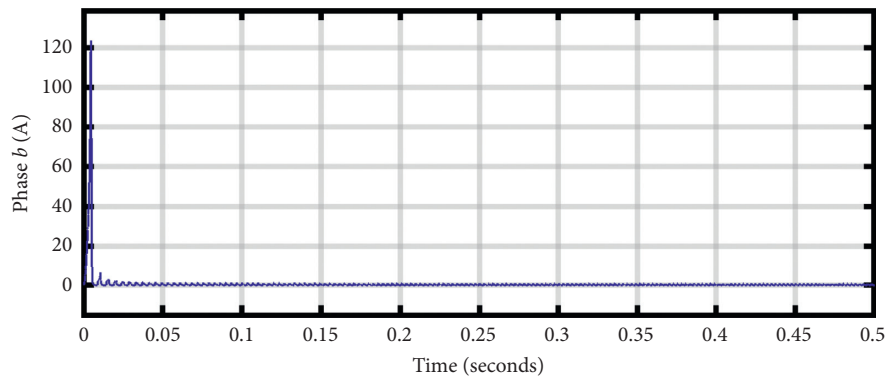FIGURE 48: Phase *a* of an improved model of SRM.

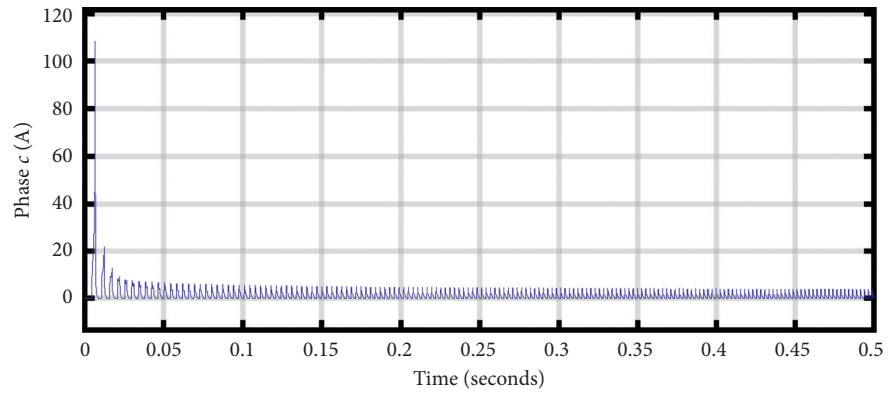
FIGURE 49: Phase *b* of an improved model of SRM.

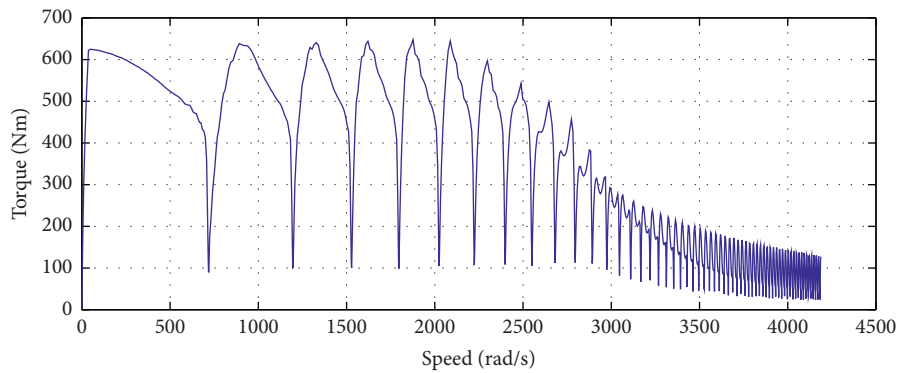FIGURE 50: Phase *c* of an improved model of SRM.

FIGURE 51: Graph between torque and speed of SRM without the proposed technique.
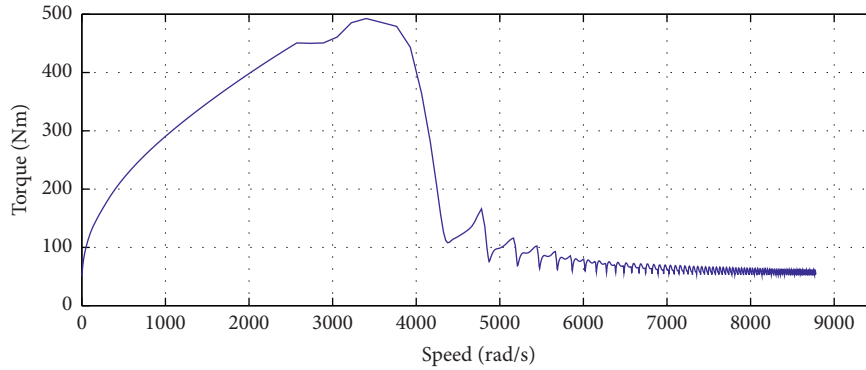
FIGURE 52: Graph between torque and speed of SRM with the proposed technique.

TABLE 20: Torque ripple improvement.

Sr. no.	Case	Torque ripple	Torque improvement (%)
1	Simplified model	12.61	—
2	ANN-based speed and current simple	9.73	22
3	ANN-based current and speed simple	8.53	31.9
4	ANN-based speed with noise and current simple	9.66	23.4
5	ANN-based current with noise and speed simple	8.469	32
6	Improved model	7.49	40.6

TABLE 21: Comparison of the proposed technique with existing techniques.

Parameters	Proposed technique	PI controller	Simplified model of SRM
MSE	MSE is very much reduced in ANN	MSE is higher as compared with ANN	MSE is of large value as compared with ANN and PI
Accuracy	Improved accuracy	Less accurate in comparison with ANN	Lack of accuracy
Conversions	Speedily conversions with reduction in torque ripple	Slow conversions as compared with ANN	Slow conversions as compared with ANN
Generalization	Good generalization when retrained improved performance with time	No generalization property has to be returned and no improvements with time	No generalization property
Simulation speed	Time taken for computation is less and fast in operation	Long computing time	Slow
Performance	Better performance for nonlinear characteristics of SRM	Performance is not up to the expected level	Poor performance
Regression	Regression is in the proximity of 1 which validate performance and fast change of parameters in ANN	No regression property, SRM demands a fast change of parameters, and the PI controller has a slow change of parameters	Nil
Complexity	Less complex because no mathematical model included	Complex because mathematical model included	Complex than ANN and PI controller
Stability	Good stability	Less stable than ANN	Less stable

The waveform of instantaneous torque with and without the proposed method at different reference speed is displayed in Figures 51 and 52.

Table 20 shows torque ripple value and percentage value of torque improvement in comparison with the simplified model.

6. Effectiveness of the Proposed Technique

The effectiveness of this proposed technique based on ANN for SRM is further elaborated by comparing it with other existing techniques. A comparison is drawn in Table 21.

7. Conclusion

Simulation results validate that waveforms of torque, speed, and three-phase currents are worse in the case of a simplified model of SRM and show poor performance. It is also proved from graphical results that mean square error (MSE), calculated for speed and current control, is significantly reduced with ANN. In other words, an improved model of ANN-based SRM shows a noticeable reduction in torque ripples, resulting in performance improvement. Torque, speed, and current are the performance parameters and are improved through ANN. Therefore, simulation results verify the effectiveness and validity of an improved model of SRM

and capability of reduction of torque ripples through the ANN algorithm. This leads to the deployment of SRM in various useful applications of everyday life because of an increase in efficiency, ruggedness, and employability in a dynamic environment. High-speed applications can be controlled (stopped and loaded) in a fine way. High-speed motors have benefits of high power density which is an important parameter of traction motors in electric vehicles. Therefore, high-speed SRM is the best alternative for this application because of its stability, reduced dependability, and the simplest structure. Moreover, ANN shows better results for SRM because it has fast convergence, robust, stable, and best suited for nonlinear problems.

Data Availability

No data were used to support the findings of this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the publication of this paper.

Authors' Contributions

Raheel Muzzammel conceptualized the study. Iqra Tariq and Raheel Muzzammel were responsible for methodology. Iqra Tariq provided software. Raheel Muzzammel validated the study. Iqra Tariq was responsible for formal analysis. Iqra Tariq investigated the study. Iqra Tariq, Raheel Muzzammel, Umar Alqasmi, and Ali Raza were responsible for resources, data curation, and writing—review and editing: . Iqra Tariq contributed to writing—original draft preparation. Raheel Muzzammel supervised the study. Raheel Muzzammel was responsible for project administration. Raheel Muzzammel, Umar Alqasmi, and Ali Raza acquired funding.

References

[1] S. Song and W. Liu, "A comparative study on modeling methods for switched reluctance machines," *Computer and Information Science*, vol. 3, no. 2, 2010.

[2] E. M. Barhoumi, A. G. Abo-Khalil, Y. Berrouche, and F. Wurtz, "Analysis and comparison of end effects in linear switched reluctance and hybrid motors," *Journal of Electrical Engineering*, vol. 68, pp. 138–142, 2017.

[3] J. F. Pan, Y. Zou, G. Cao, N. C. Cheung, and B. Zhang, "High-precision dual-loop position control of an asymmetric bilateral linear hybrid switched reluctance motor," *IEEE Transactions on Magnetics*, vol. 51, no. 11, pp. 1–5, 2015.

[4] R. Krishnan, *Switched Reluctance Motor Drives: Modelling, Simulation, Analysis, Design and Applications*, J. Irwin, Ed., CRC Press, Boca Raton, Fl, USA, 2001.

[5] T. J. E. Miller, *Electronic Control of Switched Reluctance Machines*, Elsevier, Amsterdam, Netherlands, 2001.

[6] I. Husain, "Minimization of torque ripple in SRM drives," *IEEE Transactions on Industrial Electronics*, vol. 49, no. 1, pp. 28–39, Feb 2002.

[7] Z. Xu, D. Lee, and J. Ahn, "Analysis and control of a novel bearingless switched reluctance motor with hybrid stator poles," in *Proceedings of the 2013 IEEE International Conference on Industrial Technology (ICIT)*, Cape Town, South Africa, 2013.

[8] T. J. E. Miller, "Optimal design of switched reluctance motors," *IEEE Transactions on Industrial Electronics*, vol. 49, no. 1, pp. 15–27, 2002.

[9] E. F. I. Raj and V. Kamaraj, "Neural network based control for switched reluctance motor drive," in *Proceedings of the 2013 IEEE International Conference ON Emerging Trends in Computing, Communication and Nanotechnology (ICECCN)*, Tirunelveli, India, 2013.

[10] X. Gao, X. Wang, Z. Li, and Y. Zhou, "A review of torque ripple control strategies of switched reluctance motor," *International Journal of Control and Automation*, vol. 8, no. 4, pp. 103–116, 2015.

[11] S. A. Nasar, "DC-switched reluctance motor," *Proceedings of the Institution of Electrical Engineers*, vol. 116, no. 6, pp. 1048-1049, 1969.

[12] R. B. I. Doncker and R. W. De, "DITC-direct instantaneous torque control of switched reluctance drives," in *Proceedings of the Conference Record of the 2002 IEEE Industry Applications Conference, 37th IAS Annual Meeting*, Pittsburgh, PA, USA, 2002.

[13] H. J. Brauer, M. D. Hennen, and R. W. De Doncker, "Multiphase torque-sharing concepts of predictive PWM-DITC for SRM," in *Proceedings of the 2007 7th International Conference on Power Electronics and Drive Systems*, Bangkok, Thailand, 2007.

[14] G. Shahgholian, A. R. Sahafi, and J. Faiz, "Torque ripple reduction in switched reluctance motors–a review," *Electromotion*, vol. 22, pp. 35-36, 2015.

[15] T. Husain, A. Elrayyah, Y. Sozer, and I. Husain, "Unified control for switched reluctance motors for wide speed operation," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 5, pp. 3401–3411, May 2019.

[16] S. Kumari and S. Ashok, "Design of controller for speed regulation and reduction of torque ripple in 6/4 switched reluctance motor," in *Proceedings of the 2017 IEEE International Conference on Intelligent Techniques in Control, Optimization and Signal Processing (INCOS)*, Srivilliputhur, India, 2017.

[17] G. Singh and B. Singh, "Performance improvement of switched reluctance motor drive for light electric vehicle using fuzzy logic controller," in *Proceedings of the 2018 IEEE International Conference on Power Electronics, Drives and Energy Systems (PEDES)*, Chennai, India, 2018.

[18] Z. Lin, D. S. Reay, B. W. Williams, and X. He, "Torque ripple reduction in switched reluctance motor drives using B-spline neural networks," *IEEE Transactions on Industry Applications*, vol. 42, no. 6, pp. 1445–1453, 2006.

[19] B. S. Ali, H. M. Hasanien, and Y. Galal, "Speed control of switched reluctance motor using artificial neural network controller," *Computational Intelligence and Information Technology. Communications in Computer and Information Science*, vol. 250, pp. 6–14, 2011.

[20] C. Sunita and M. V. Srikanth, "Modeling and analysis of 6/4 switched reluctance motor with torque ripple reduction," *IOSR Journal of Electrical and Electronics Engineering (IOSR-JEEE)*, vol. 7, pp. 37–42.

[21] E. Gouda, M. Hamouda, and A. R. A. Amin, "Artificial intelligence based torque ripple minimization of switched reluctance motor drives," pp. 943–948, 2016.

[22] V. Pushparajesh, M. Balamurugan, and N. Ramaiah, "Artificial neural network based direct torque control of four phase switched reluctance motor," *SSRN Electronic Journal*, 2019.

[23] A. Kailash, *Everything You need to Know about Neural Networks*, Mate Labs, 2017, https://hackernoon.com/everything-you-need-to-know-about-neural-networks-8988c3ee4491.

[24] D. Lu and N. C. Kar, "Neural network based torque control of switched reluctance motor for hybrid electric vehicle propulsion at low speeds," in *Proceedings of the 2009 IEEE International Conference on Electro/Information Technology*, Windsor, Canada, 2009.

[25] H. Yu, "Levenberg–Marquardt training," *Journal of Intelligent Systems*, vol. 16.

[26] R. Muzzammel, O. A. Sajid, M. Manzoor, A. Qayyum, and A. Shahzad, "Simulation analysis of PI based switched reluctance motor," *Journal of Engineering Research and Reports*, vol. 3, no. 2, pp. 1–12, 2019.

[27] H. E. Akhter, V. K. Sharma, A. Chandra, and K. Al-Haddad, "Determination of optimum switching angles for speed control of switched reluctance motor drive system," *Indian Journal of Engineering and Materials Sciences*, vol. 11, pp. 151–168, 2004.

[28] M. Nashed, S. Mahmoud, M. El-Sherif, and E. Sami, "Optimum change of switching angles on switched reluctance motor performance," *International Journal of Current Engineering and Technology*, vol. 4, no. 2, 2014.

[29] J. Deskur, T. Pajchrowski, and K. Zawirski, "Optimal control of current commutation of high speed SRM drive," in *Proceedings of the 13th International Power Electronics and Motion Control Conference*, pp. 1204–1208, Poznan, Poland, September, 2008.

[30] C. Mademlis and I. Kioskeridis, "Performance optimization in switched reluctance motor drives with online commutation angle control," *IEEE Transactions on Energy Conversion*, vol. 18, no. 3, pp. 448–457, 2003.

[31] A. Sharma, "What is negative speed?," 2020, https://www.quora.com/what-is-negative-speed.