sciendo
Power Electronics and Drives
Volume 8(43), 2023 DOI: 10.2478/pead-2023-0021

Rotor Speed and Load Torque Estimations of Induction Motors via LSTM Network

Research paper

Mehmet Muzaffer Kosten[1,2], Alper Emlek[1], Recep Yildiz[1,*], Murat Barut[1]

[1]*Department of Electrical and Electronic Engineering, Nigde Omer Halisdemir University, Nigde, Türkiye*
[2]*Koç Bilgi ve Savunma Teknolojileri A.Ş., Middle East Technical University Technopolis, Ankara, Türkiye*

Received: 14 June, 2023; Accepted: 02 September, 2023

Abstract: In this study, a long short-term memory (LSTM) based estimator using rotating axis components of the stator voltages and currents as inputs is designed to perform estimations of rotor mechanical speed and load torque values of the induction motor (IM) for electrical vehicle (EV) applications. For this aim, first of all, an indirect vector controlled IM drive is implemented in simulation to collect both training and test datasets. After the initial training, a fine-tuning process is applied to increase the robustness of the proposed LSTM network. Furthermore, the LSTM parameters, layer size, and hidden size are also optimised to increase the estimation performance. The proposed LSTM network is tested under two different challenging scenarios including the operation of the IM with linear and step-like load torque changes in a single direction and in both directions. To force the proposed LSTM network, it is also tested under the variation of stator and rotor resistances for the both-direction scenario. The obtained results confirm the highly satisfactory estimation performance of the proposed LSTM network and its applicability for the EV applications of the IMs.

Keywords: *long short-term memory • deep neural network • electrical vehicle • induction motor • state and parameter estimation*

1. Introduction

Climate change/global warming is one of the most dangerous threats to humanity, even to life on the planet. The main reason for climate change is the greenhouse effect, which results in heating up of the planet's surface. Owing to human activities such as burning fossil fuels, the emission of carbon-dioxide, the main component of greenhouse gases, increases, which causes strengthening of the greenhouse effect. Therefore, attempts are being made to ensure that precautions are in place to reduce carbon-dioxide emissions all over the world and also to save humanity's limited natural resources. From this point of view, electrical vehicles (EVs), using an electric motor instead of internal combustion engines, have gained importance and become more popular. As in the industry, this tendency towards EVs reveals the importance of the efficiency and robustness of the drive systems. There are some types of electric motor that are commonly used in EVs, and the induction motor (IM) is one of them. Some reasons for using IMs in both industrial and EV applications are reasonable cost, simplicity in construction, working in harsh environments, and fewer maintenance requirements. There are also some control strategies used in the IM drives to perform the high-performance control application of IMs, such as direct/indirect vector control (VC) (Reddy et al., 2022; Vas, 1998), direct torque control (Alsofyani and Idris, 2016; Demir, 2023; Vas, 1998), and model predictive control (Rodriguez et al., 2012; Wróbel et al., 2020). However, the highly nonlinear model of the IM whose parameters change with operating conditions makes high-performance control challenging. To overcome this challenge, there are different model-based approaches proposed in the literature, such as model reference adaptive system (MRAS) (Bednarz and Dybkowski, 2019), full/reduced order observers (Vicente et al., 2010), sliding mode observers (SMO) (Zhang et al., 2020), Luenberger observers (Yin et al., 2021), and nonlinear versions of the Kalman filter (Yildiz et al., 2020a, 2023).

* *Email: ryildiz@ohu.edu.tr*

Along with the model-based methods, there are also neural network (NN) based solutions in the literature. In Simoes and Bose (1995), an NN estimator using stator currents and fluxes is designed to obtain rotor flux, electromagnetic torque, and unit vectors ($\cos\theta_e$ and $\sin\theta_e$) to perform direct VC of IMs. Furthermore, the estimation performance of the proposed NN estimator is compared to that of voltage model flux estimator in simulations. It is stated in the paper that, compared to the voltage model flux estimator, the NN has distinct advantages, such as fault tolerance, fast execution speed, learning input–output relation, and harmonic immunity. In Kim et al. (2001), a MRAS-based NN speed estimator is proposed to estimate rotor speed of the IM in real-time experiments. In order to obtain rotor fluxes utilised in the NN speed estimator, the voltage and current models are used as the reference and adjustable models, respectively. In Karanayil et al. (2005) the online estimation of stator and rotor resistances is performed by using fuzzy logic based estimator and NN-based estimator, respectively. Here, a programmable cascade filter based estimator is utilised to obtain flux estimation that is required for rotor resistance estimation. Contrary to the study realized by Karanayil et al. (2005), Karanayil et al. (2007) perform the online estimation of stator and rotor resistances in speed-sensorless case. In Orlowska-Kowalska et al. (2010), adaptive sliding mode neuro-fuzzy controller is used in the speed-sensorless VC of the IM. The rotor speed estimation is performed by using stator current based MRAS. In Sun et al. (2013), the NN inverse method based speed observation is proposed for speed-sensorless VC of the IM. In Imane et al. (2017), a neural adaptive Kalman filter is proposed for VC of the IM in simulations. The stator currents and rotor fluxes are estimated by Kalman filter, and the rotor speed is obtained by adaptive linear neuron (ADALINE) NN based adaptation mechanism. In El Merrassi et al. (2021), a neuro-MRAS observer is designed to obtain the rotor flux and speed for speed-sensorless VC of the IM. The proposed observer is composed of changing the adaptive model in the MRAS by two-layer NN and tested in simulations.

Considering the EVs applications, since EVs have processing units with a high capability, it is also possible for the researcher to use deep neural network (DNN)-based solutions in high-performance control applications of the IMs. As for the DNN-based studies, the IM dynamics are obtained by an encoder–decoder network with recurrent skip connection in Verma et al. (2020a). In the feature extraction part of the network, the one-dimensional (1-D) encoder–decoder network is used for good learning capacity compared to recurrent neural network (RNN) and long short-term memory (LSTM). It is stated in the paper that, since the convolutional networks do not learn temporal relationships, a skip connection layer including RNN is added to the encoder–decoder network. In Verma et al. (2020b), different DNNs are compared in terms of both machine learning and electrical engineering performance metrics in simulations. Here, the inputs of the DNNs are the rotating axis components of stator voltages and currents while the outputs are the rotor speed and electromagnetic torque. It is emphasised that although both machine learning and electrical engineering metrics point out the same methods, electrical engineering metrics are more suitable in terms of different DNN evaluation merits. In Acikgoz and Korkmaz (2021), a deep learning method based on the LSTM is presented to estimate the rotor speed of the IM in simulations. The input parameters of the LSTM network are the rotating axis components of stator voltages and currents, as in Verma et al. (2020b). The effect of the increasing hidden neuron number on the training process is also examined and the optimal neuron number is determined for simulations. Moreover, four performance metrics, the correlation coefficient (R), the R-squared (R^2), the mean absolute error (MAE), and the root mean square error (RMSE), are used to evaluate the estimation performance of the LSTM-based speed observer. Both the training and test data include only single direction of rotation and the load torque is applied as a constant. In Ilten et al. (2022), the LSTM-based DNN method is designed to perform the rotor speed estimation of the IM by using the stationary axis components of measured stator voltages and currents in real-time experiments. To determine the optimal parameters of the LSTM network, different performance metrics such as mean square error (MSE), RMSE, MAE, and root relative squared error (RRSE) are used. Furthermore, the estimation performance of the LSTM-based speed observer using the RMSE metric, the most successful one, is compared to that of an SMO that performs the speed estimation. However, reviewing the experimental results, the LSTM-based observer is tested under the condition of a single direction of rotation, as in Acikgoz and Korkmaz (2021). In Verma et al. (2023), the rotor speed and electromagnetic torque are estimated in simulations and real-time experiments. Since the measured input and output data are noisy in real-time, a denoising method called meta denoiser (MD) is proposed. For the simulation case, as opposed to Verma et al. (2020b), the noisy condition is also tested by noise modelling and thus the MD is also tested under the noisy simulation case. The proposed method (DiagBiRNN) in Verma et al. (2020b) is improved by using the MD. The results demonstrate that the effect of the MD increases the estimation performance of the DiagBiRNN. In Teler and Orłowska-Kowalska (2023), the stator currents of the IM are estimated by the LSTM network under different fault

conditions in simulations. Furthermore, to test the robustness of the LSTM network, the parameter variations are also examined.

In this paper, an LSTM network based rotor mechanical speed and load torque estimator is designed for the IM drive to be used in EVs applications. The proposed LSTM network uses rotating axis components of stator voltages and currents while the output of the network is the rotor mechanical speed and load torque. To determine the cost-effective and most accurate architecture of the LSTM network, different number layers and hidden sizes are examined and compared. One of the reasons why the present research might be considered relatively unique is that it involves the evaluation of both different number layers and different hidden sizes over the estimations. Compared to Acikgoz and Korkmaz (2021) and Ilten et al. (2022), both the train data and test data contain an operation condition in both directions of rotation along with the challenging load torque variations. Therefore, the operation of the IM in both directions is examined in this paper. Finally, the effects of the parameter variations referring to rotor and stator resistances on the proposed LSTM network are also investigated.

This paper consists of five sections. While the first section has provided a detailed review of the literature pertaining to this topic, the remainder of this paper is organised as follows: Section 2 presents the details of the dynamic IM model and the indirect vector controlled IM drive; Section 3 explains the LSTM-based rotor mechanical speed and load torque estimation method; and Sections 4 and 5, respectively, provide the simulation studies and conclusion.

2. Dynamic Model and Indirect VC of the IM

The dynamic model of the IM can be defined by a fifth-order mathematical model with both stator flux and rotor flux. Both stator and rotor flux-based models can be obtained at stator stationary axis ($\alpha\beta$ –) and rotating axis (dq –). The first four equations of the dynamic IM model refer to the electrical subsystem while the last equation represents the mechanical subsystem obtained by adaptation of Newton's second law of motion, namely the equation of motion, for the rotational systems. The rotor flux-based fifth-order dynamic model of the IM is presented in Eqs. (1)–(5) as in Yildiz et al. (2020a).

$$\frac{di_{sd}}{dt} = -\left(\frac{R_s}{L_\sigma} + \frac{L_m^2 R_r}{L_r^2 L_\sigma}\right) i_{sd} + \omega_s i_{sq} + \frac{L_m R_r}{L_r^2 L_\sigma}\phi_{rd} + \frac{L_m p_p \omega_m}{L_\sigma L_r}\phi_{rq} + \frac{1}{L_\sigma} v_{sd} \tag{1}$$

$$\frac{di_{sq}}{dt} = -\omega_s i_{sd} - \left(\frac{R_s}{L_\sigma} + \frac{L_m^2 R_r}{L_r^2 L_\sigma}\right) i_{sq} - \frac{L_m p_p \omega_m}{L_\sigma L_r}\phi_{rd} + \frac{L_m R_r}{L_r^2 L_\sigma}\phi_{rq} + \frac{1}{L_\sigma} v_{sq} \tag{2}$$

$$\frac{d\phi_{rd}}{dt} = \frac{R_r L_m}{L_r} i_{sd} - \frac{R_r}{L_r}\phi_{rd} + \omega_r \phi_{rq} \tag{3}$$

$$\frac{d\phi_{rq}}{dt} = \frac{R_r L_m}{L_r} i_{sq} - \omega_r \phi_{rd} - \frac{R_r}{L_r}\phi_{rq} \tag{4}$$

$$\frac{d\omega_m}{dt} = \frac{3 p_p L_m}{2 J_T L_r}(\phi_{rd} i_{sq} - \phi_{rq} i_{sd}) - \frac{B_T}{J_T}\omega_m - \frac{t_L}{J_T} \tag{5}$$

where v_{sd} and v_{sq} are the dq – axis components of stator voltages; i_{sd} and i_{sq} are the dq – axis components of stator currents; ϕ_{rd} and ϕ_{rq} are the dq – axis components of rotor fluxes; L_s and L_r are the stator and rotor inductances, respectively; L_m is the magnetising inductance; R_s and R_r are the stator and rotor resistances, respectively; $L_\sigma = \sigma L_s = L_s - L_m^2/L_r$ is the stator transient inductance; ω_s is the synchronous speed, ω_r is the slip speed; ω_m is the rotor mechanical speed; t_L is the load torque; and B_T and J_T are viscous friction term and the total inertia of the system, respectively.

Since, in this paper, only ω_m and t_L estimations are performed by the proposed LSTM network without flux measurement/estimation, the indirect VC method is used as the IM drive. The block diagram of the indirect vector controlled IM drive implemented in Matlab/Simulink is presented in Figure 1 (Vas, 1998). In Figure 1, the proportional-integral (PI) controller whose coefficients are determined by a trial-and-error method is used in speed

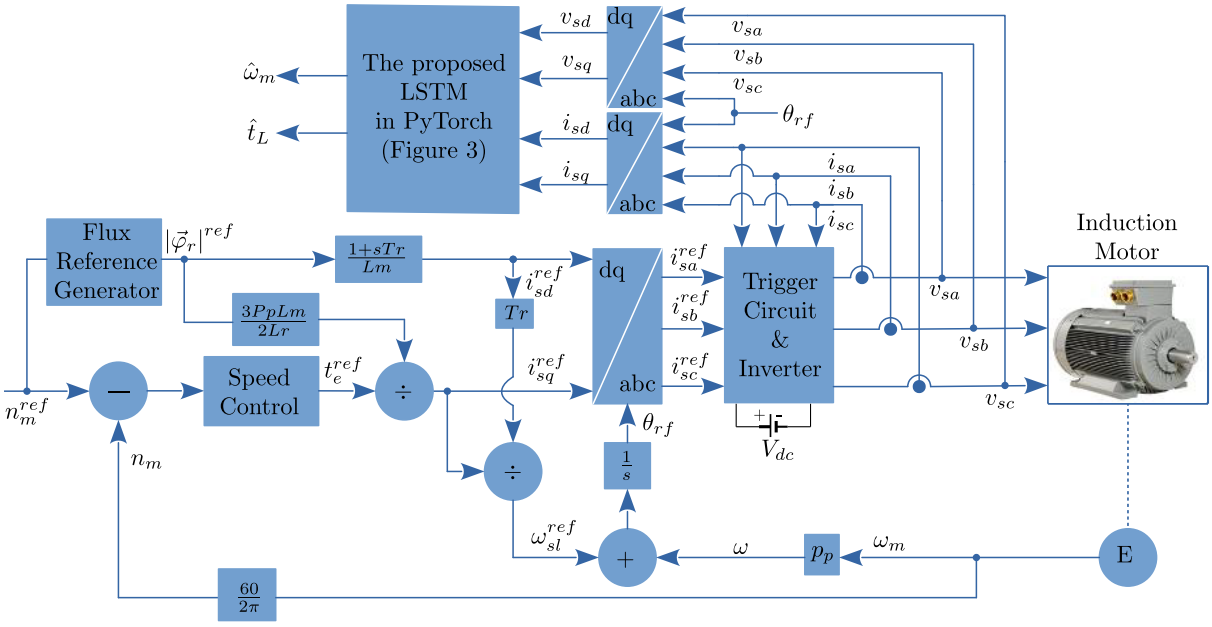

Fig. 1. Indirect vector controlled IM drive system (Vas, 1998). IM, induction motor.

$P[kW]$	$V[V]$	$I[A]$	$f[Hz]$	$n_m[rad/\min]$	$t_L[Nm]$
3	380	6.9	50	1,430	20
p_p	$R_s[\Omega]$	$R_r[\Omega]$	$L_s[H]$	$L_r[H]$	$L_m[H]$
2	2.283	2.133	0.2311	0.2311	0.22

IM, induction motor.

Table 1. Rated parameter values and specifications of the IM.

control loop. A reference-speed dependent flux reference generator, which is detailed in Yildiz et al. (2020b), is used to obtain the flux reference of the IM drive. Moreover, as is well-known for the indirect VC, the summation of the slip frequency reference (ω_{sl}^{ref}) and the rotor electrical frequency (ω) is utilised to obtain rotor flux angle (θ_{rf}), as indicated in Figure 1.

In Figure 1, s represents the Laplace operator, and $T_r = L_r/R_r$ refers to the rotor time constant. The rated parameter values and specifications of the IM in Figure 1 are given in Table 1 (Yildiz et al., 2020a, 2023).

3. The LSTM-based Speed and Torque Estimation Model

The LSTM is a kind of RNN that can learn long-term dependencies in sequential data. LSTMs can process sequential information better than other types of NNs, as they have the ability to selectively remember and forget information from previous time steps. This allows them to preserve information for longer periods of time and avoid the vanishing gradient problem, making them a popular choice for many machine learning applications such as natural language processing (Sundermeyer et al., 2015), speech recognition (Ullah et al., 2018), and other time-series applications (Selvin et al., 2017).

The LSTM architecture has a memory cell that can add or remove information at each time step. There are four fundamental components involved in this process: input gate, forget gate, output gate, and cell state, which is also known as the state candidate gate or the input modulation gate. The block diagram of a typical LSTM cell is given in Figure 2.

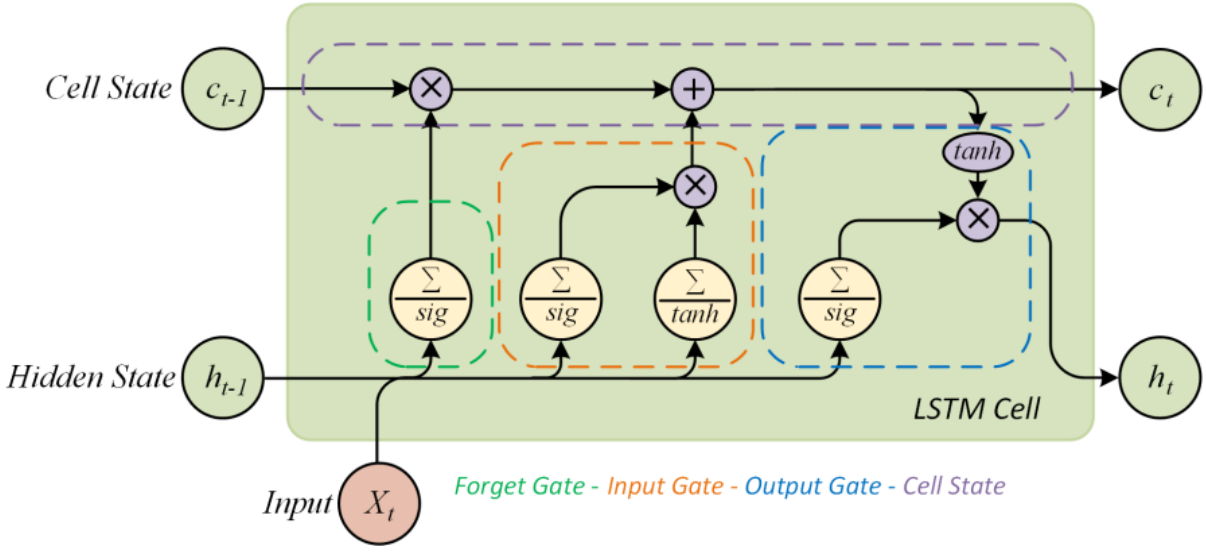

Fig. 2. The basic LSTM cell. LSTM, long short-term memory.

As for the fundamental steps of the LSTM cell, the forget gate, the first step of LSTM, decides whether the information from the previous cell will be remembered or forgotten. The decision process is performed by comparing the information from the previous cell (h_{t-1}) and the current cell (x_t). The final decision is made by applying the comparison results to the sigmoid function whose output can be between 0 and 1. At the end of the decision process, information with 0 is eliminated, and information with 1 is added to the cell state to use. The mathematical expression of the forget gate is given in Eq. (6).

$$i_t = \sigma\left(W_{if}x_t + b_{if} + W_{hf}h_{t-1} + b_{hf}\right) \tag{6}$$

In Eq. (6), W represents the weight matrices and b represents the bias term. Furthermore, the sigmoid function is represented by σ.

The second step of the LSTM is the input gate, which is responsible for regulating the flow of information into the LSTM cell. The input gate is to decide whether current information will be stored in the cell state to predict the next output. For this purpose, x_t and h_{t-1} are passed through both a sigmoid layer and a tanh layer. Similar to the case in forget gate, the sigmoid function creates output values between 0 and 1 according to the importance of the input values. Moreover, the tanh layer creates a vector called candidate vector (g_t) whose values lie between –1 and 1 for helping to regulate the network. The mathematical operations performed in the input gate are given in Eqs. (7) and (8).

$$f_t = \sigma\left(W_{ii}x_t + b_{ii} + W_{hi}h_{t-1} + b_{hi}\right) \tag{7}$$

$$g_t = \tanh\left(W_{ig}x_t + b_{ig} + W_{hg}h_{t-1} + b_{hg}\right) \tag{8}$$

The third part of the LSTM cell, the cell state, is one of the most important components of the LSTM. The cell state holds long-term information that the network needs to learn. In the cell state, the LSTM network learns long-term dependencies by updating and storing this information, while the cell state processes inputs and retains historical information in its memory. The mathematical expression for the output of the cell state (c_t) is given in Eq. (9).

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \tag{9}$$

The output gate, the last part of the LSTM cell, determines the next hidden state (h_t) of the cell. The next hidden state (h_t) is calculated by using the information obtained at the cell state (c_t) and the output of the sigmoid function

whose inputs are x_t and h_{t-1}. The mathematical operations performed in the output gate are given in Eqs. (10) and (11).

$$o_t = \sigma\left(W_{io}x_t + b_{io} + W_{ho}h_{t-1} + b_{ho}\right) \tag{10}$$

$$h_t = \mathrm{o_t} \odot \tanh(c_t) \tag{11}$$

When the dynamic model in Eqs. (1)–(5) of the IM is considered, it has a fifth-order nonlinear model, which contains the multiplications of the state variables. By applying the forward Euler approximation in Eq. (12) to Eqs. (1)–(5), the discretised rotor flux-based IM model can be obtained. In the discrete IM model, as is well-known, the state variable value at the instant $k+1$ is dependent on its previous value at time k. Furthermore, the state variables (i_{sd}, i_{sq}, ϕ_{rd}, ϕ_{rq}, and ω_m) of the IM model can be controlled by the control inputs, v_{sd} and v_{sq}. Therefore, by using information pertaining to v_{sd}, $_{sq}$, i_{sd}, and i_{sq}, it is possible for the proposed LSTM network to construct the input-output relationship between the measured v_{sd}, v_{sq}, i_{sd}, and i_{sq} inputs and ω_m and t_L outputs. The proposed LSTM network structure used for the estimation of ω_m and t_L is given in Figure 3. This network structure consists of four layers, namely input layer, LSTM layer, fully connected layer, and output layer, sequentially.

$$x_t = \frac{x_{k+1} - x_k}{T} \tag{12}$$

In the input layer, before the measured v_{sd}, v_{sq}, i_{sd}, and i_{sq} inputs are applied to the proposed LSTM network, these values are normalised by using the min–max limit of each input to the [0,1] range to reduce dependencies and increase the prediction accuracy of ω_m and t_L. As is well-known, the feature extraction process is more challenging when the $\alpha\beta-$ axis components of the stator voltages ($v_{s\alpha}$ and $v_{s\beta}$) and currents ($i_{s\alpha}$ and $i_{s\beta}$) are used as inputs owing to the sinusoidal variations of the input data. Therefore, considering the sequential information at input data, it is more meaningful to perform feature extraction from the input data in the $dq-$ axis. That is why this study is realised by using $dq-$ components of stator voltages (v_{sd} and v_{sq}) and currents (i_{sd} and i_{sq}). Here, the corresponding min–max limits for the voltages and currents are determined as [–400 V, 400 V] and [–10 A, 10 A], respectively. In the second layer, the LSTM network that takes the normalised value of v_{sd}, v_{sq}, i_{sd}, and i_{sq} as input is used. The length of the feature vectors obtained from the output of the LSTM networks depends on the number of layers in the LSTM structure. Therefore, firstly, the features obtained from the different LSTM layers are concatenated in the third layer. Then, two fully connected layers are applied to the output of the LSTM network to obtain ω_m and t_L estimations. As can be seen from Figure 3, the output size of the second fully connected layer is equal to the number of estimated values. The output layer directly gives the estimated ω_m and t_L values.

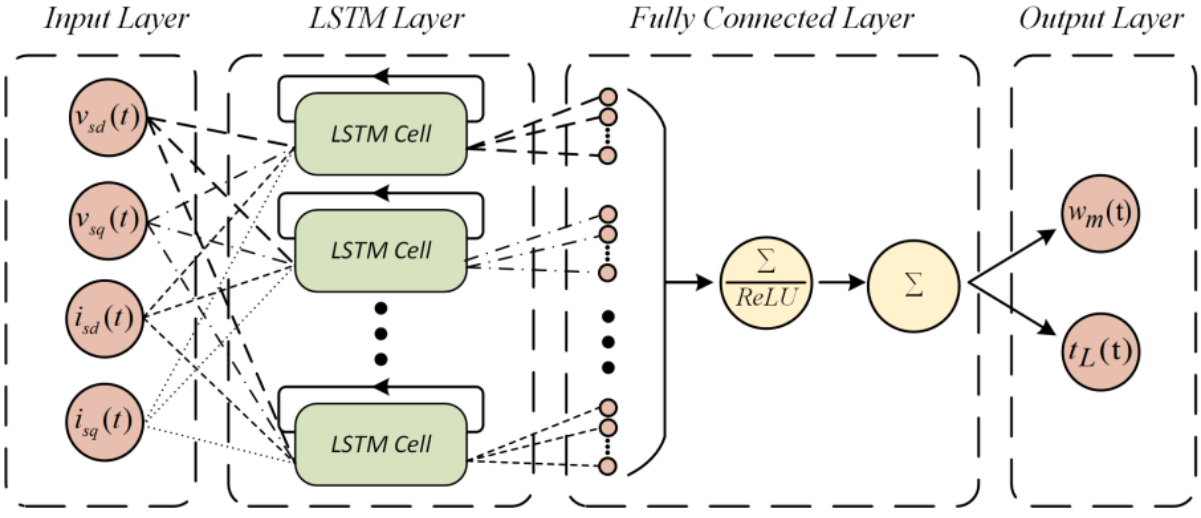

Fig. 3. LSTM-based network structure for speed and torque estimation. LSTM, long short-term memory.

4. Simulation Studies

In this section, by using the indirect vector controlled IM drive given in Figure 1, the training and test data for the proposed LSTM network are created. In order to make the training process comprehensive, two different scenarios are designed, and corresponding datasets are used in the training process. Moreover, the performance of the LSTM network used to perform estimation of both ω_m and t_L is tested under challenging scenarios that contain the ω_m, t_L, R_s, and R_r changes. To increase the estimation performance, the effects of the parameter variations of the LSTM are also examined and the optimum network parameter values are determined.

4.1. Datasets and evaluation metrics

In order to perform the training process, two different datasets that include both directions of rotation and t_L changes are generated by using the IM drive, as indicated in Figure 1. Both scenarios realised for the training contain the four quadrant operation of the IM. By generating two different datasets, the present research aims to cover, as much as possible, the operation conditions of IM in the training process. The training datasets are referred to as $Train\_A$ and $Train\_B$, and these datasets are presented in Figures 4(a) and (b), respectively.

To test the estimation performance of the proposed network, two datasets that include the operation of the IM along with t_L changes in a single direction (Figure 5(a)) or in both directions (Figure 5(b)) are generated, as seen in Figure 5. The single-direction scenario contains load torque variations while the rotor speed is positive and zero, as indicated in Figure 5(a). The both-direction scenario includes load torque changes for each direction of rotation, as shown in Figure 5(b). Furthermore, the effect of the parameter variations over the estimation performance of the

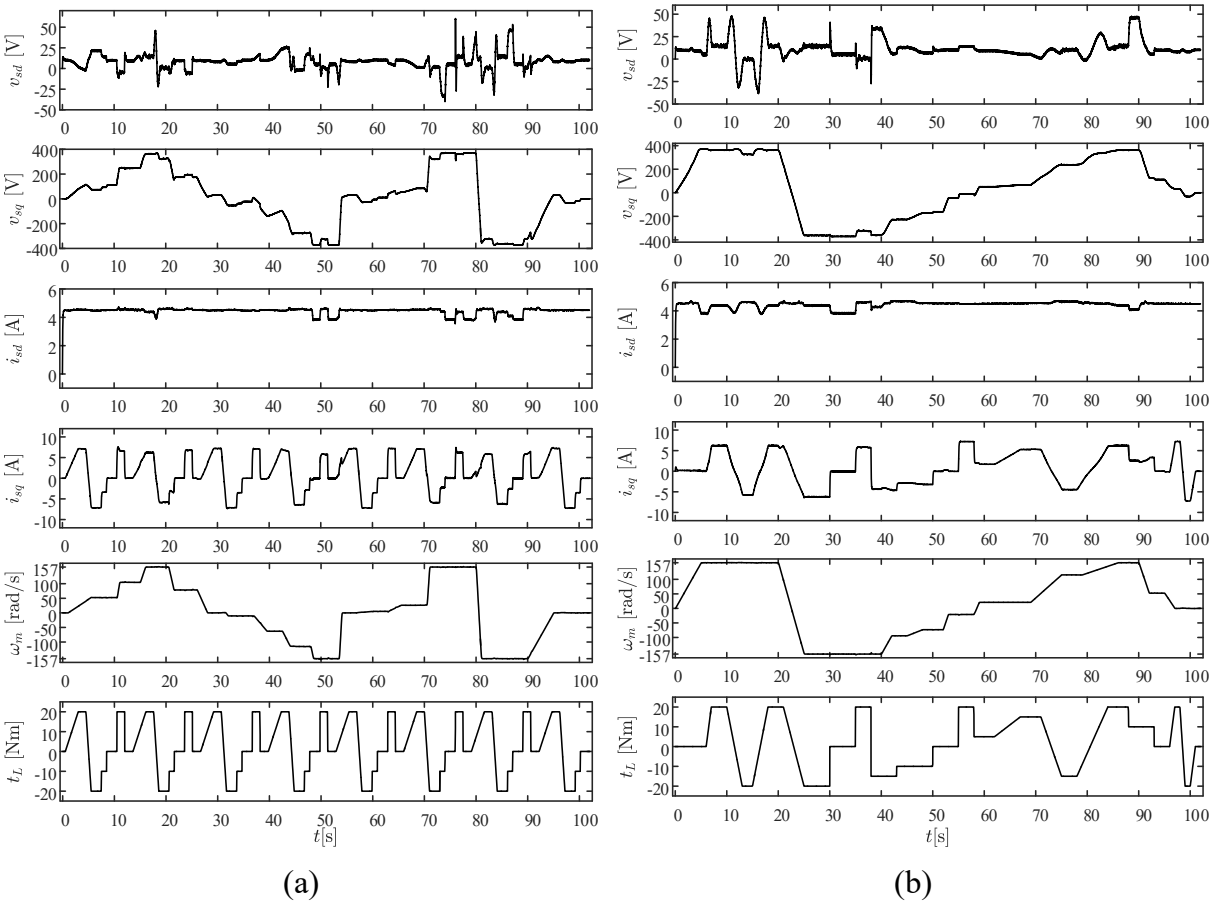

Fig. 4. a) *Train_A* dataset and (b) *Train_B* datasets obtained by using the IM drive in Figure 1 for the training process of the proposed LSTM network. IM, induction motor; LSTM, long short-term memory.

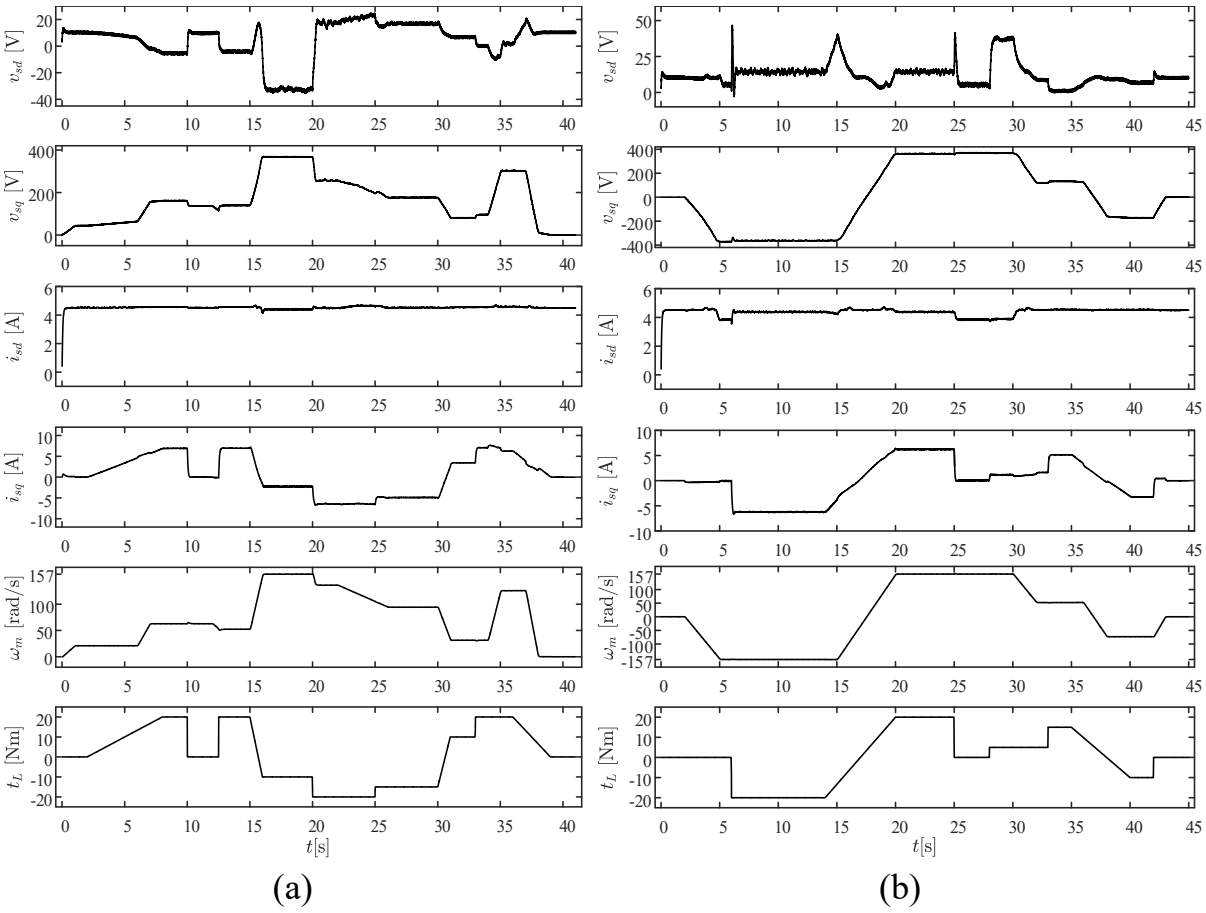

Fig. 5. Test data for operation of the IM in (a) single direction test data and in (b) both directions test data. IM, induction motor.

proposed network is also examined by performing R_s and R_r variations in the both-direction scenario, as indicated in Figure 5(b). The obtained test results are given in the following subsections.

In the evaluation of the estimation performance of the proposed network, two commonly used metrics, RMSE and MAE, are utilised. The mathematical expressions for these metrics can be given as follows:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(x_i - \hat{y}_i\right)^2} \tag{13}$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|x_i - \hat{y}_i\right| \tag{14}$$

4.2. Implementation details

Before starting the training process, each channel of the input was normalised by a determined min–max limit. The proposed architecture is implemented by using the PyTorch (Paszke et al., 2019) framework and the RMS loss value optimised with Adam algorithm ($\beta_1 = 0.9$ and $\beta_2 = 0.9$). The training process is performed on two NVIDIA RTX 2080 Ti GPUs. As aforementioned, $Train\_A$ and $Train\_B$ datasets are used in the training of the proposed LSTM network. The datasets in a single direction and in both directions (Figure 5), which are different from the training ones, are used in the test process.

The training of the proposed network is comprised of two stages, which are the initial and fine-tuning stages. In the initial stage, all weight values of the network are trained with 100 epochs and a batch size of 1,024. The fine-tuning step is separately carried out for the outputs of the network, which are estimated values of ω_m and t_L. At this stage,

all weights determined at the initial stage are frozen except for the ones related to the relevant output in the last layer. Thus, the weights of the relevant output in the last layer of the network are only ones updated. In this way, the last layer of the network is improved in the fine-tuning stage to better reflect the features obtained in the previous layers.

The learning rate for the initial training stage is set to 0.01 and divided by 2 every 20 epochs. For the fine-tuning process carried out with 50 epochs, the learning rate is set to 0.001 and divided by 2 every 20 epochs. The final network was obtained after the fine-tuning process.

4.3. LSTM network parameter optimisation

As is well-known, the LSTM network structure has two main parameters, hidden size and layer size, which can be optimised to increase network performance. To obtain high-performance estimation of ω_m and t_L, the effects of the LSTM network parameter changes are also examined in this paper. Firstly, the layer size parameter is set to 1 and different hidden size values such as 1, 2, 4, 8, 16, 32, and 64 are tested. Since a normalisation process is applied to both the input and output data of the network, the obtained RMSE and MAE results of the network, against the changing hidden size value, are given for the normalised values in Table 2. Moreover, the presented RMSE and MAE results in Table 2 are the mean value of the corresponding errors for ω_m and t_L.

As can be seen from Table 2, the optimum hidden size value is determined as 8 for both test datasets. Having determined the optimum hidden size parameter, the subsequent task is to determine the second parameter, which is layer size. The hidden size parameter is set to its optimum value, 8, and the layer size value is changed from 1 to 4. The obtained RMSE and MAE results demonstrating the optimum layer size as 1 are presented in Table 3. Similarly to Table 2, the results given in Table 3 are obtained for the normalised values and they are the mean values regarding the errors of ω_m and t_L. Since the optimum parameter values are determined in this part, these optimum values are used in all the tests performed in the remainder of the present study.

Hidden size	Single direction dataset		Both direction dataset	
	RMSE ($\times 10^{-3}$)	MAE ($\times 10^{-3}$)	RMSE ($\times 10^{-3}$)	MAE ($\times 10^{-3}$)
1	124.454	106.827	92.582	78.517
2	10.072	8.366	9.102	7.381
4	11.322	9.374	9.390	7.409
8	7.313	5.645	7.095	5.278
16	10.049	7.875	8.930	7.093
32	9.939	7.779	8.419	6.690
64	11.562	9.641	8.829	7.279

MAE, mean absolute error; RMSE, root mean square error.

Table 2. The performance of the proposed network with different hidden size values in single-direction and both-directions datasets.

Layer size	Single direction dataset		Both direction dataset	
	RMSE ($\times 10^{-3}$)	MAE ($\times 10^{-3}$)	RMSE ($\times 10^{-3}$)	MAE ($\times 10^{-3}$)
1	7.313	5.645	7.096	5.277
2	9.237	7.158	8.775	6.422
3	9.225	7.346	8.663	6.531
4	9.106	7.304	8.609	6.489

MAE, mean absolute error; RMSE, root mean square error.

Table 3. The performance of the proposed network with different layer size values in single-direction and both-directions datasets.

4.4. Simulation results

In this part, the simulation results for the proposed LSTM network are presented for different test scenarios after the application of the de-normalisation process. In Figures 6–10, '$\hat{\cdot}$' represents the estimated values; '$\hat{\cdot}^{\mathrm{P}}$' refers to estimation with only initial training (without fine-tuning); '$\hat{\cdot}^{\mathrm{FT}}$' refers to estimation with fine-tuning; and '$e_{(.)}$' refers to estimation error calculated by the differences between the measured and estimated values. In the simulation results, firstly, to demonstrate the accuracy improvement caused by the fine-tuning process, the estimation results with and without fine-tuning are compared, and the comparison results are given in Figure 6.

Furthermore, the RMSE and MAE results for the estimations with and without fine-tuning are presented in Table 4. Figure 7 presents the estimation results of the proposed LSTM network related to the single-direction test dataset while Figure 8 shows estimation results associated with the both-directions test dataset. In order to test the estimation performance of the proposed network under parameter variations, R_s and R_r values are changed to 1.2 times higher/lower values of their rated values in the both-directions scenario, as can be seen in Figure 5(b). The estimation results of the proposed LSTM network against R_s and R_r changes are presented in Figures 9 and 10, respectively. To numerically show the estimation performance of the proposed network, the corresponding RMSE and MAE results for Figures 7–10 are presented in Table 5.

Considering the estimation results and corresponding estimation errors presented in Figures 6–10 and Tables 4 and 5, it is possible to arrive at the following inferences:

- As can be seen from Figure 6 and Table 4, the fine-tuning process improves the estimation accuracy of the proposed LSTM network in terms of both ω_m and t_L.
- Since the effects of the different hidden size and layer size parameters are examined in this paper, their determined optimum values in Tables 2 and 3 are used in the simulation-based tests. Therefore, highly

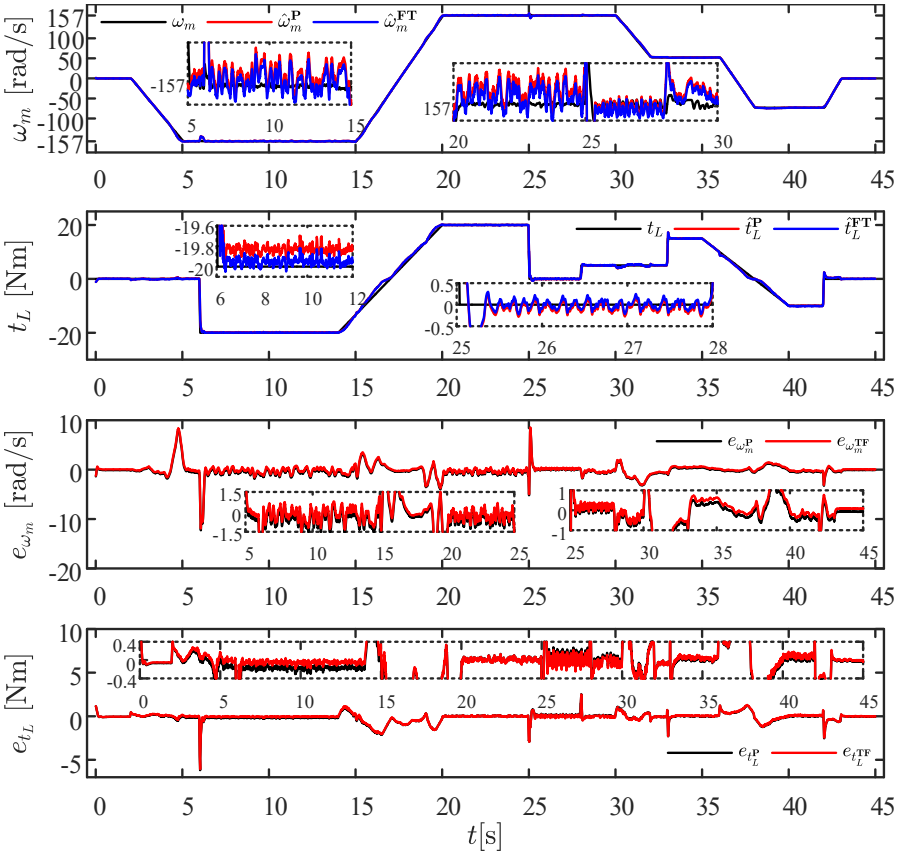

Fig. 6. Comparison between the estimations with and without fine-tuning.

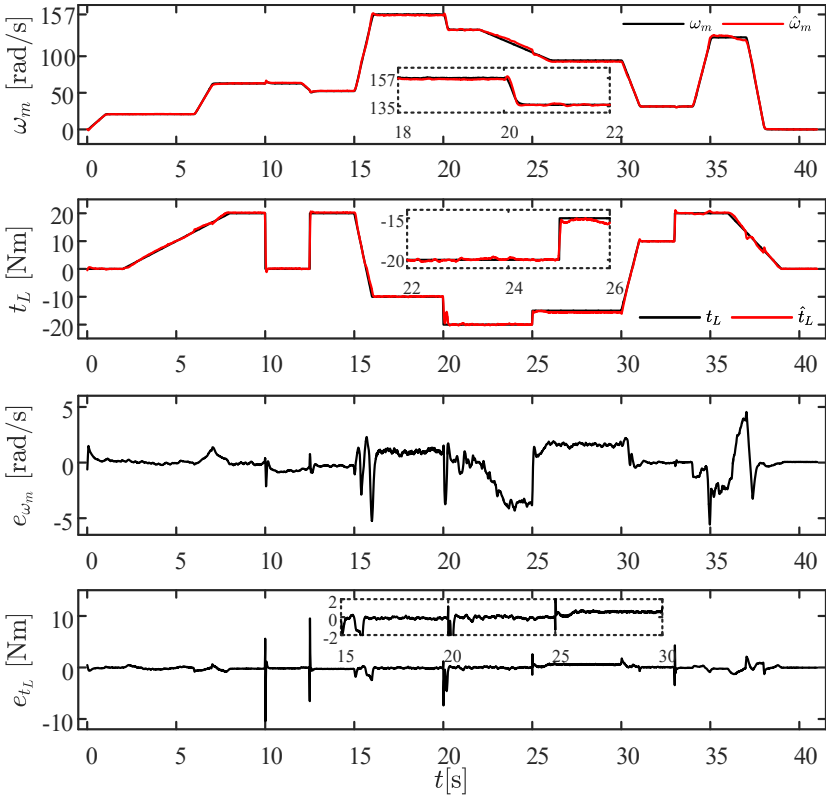

Fig. 7. Single-direction test dataset and estimation results for the proposed LSTM network. LSTM, long short-term memory.

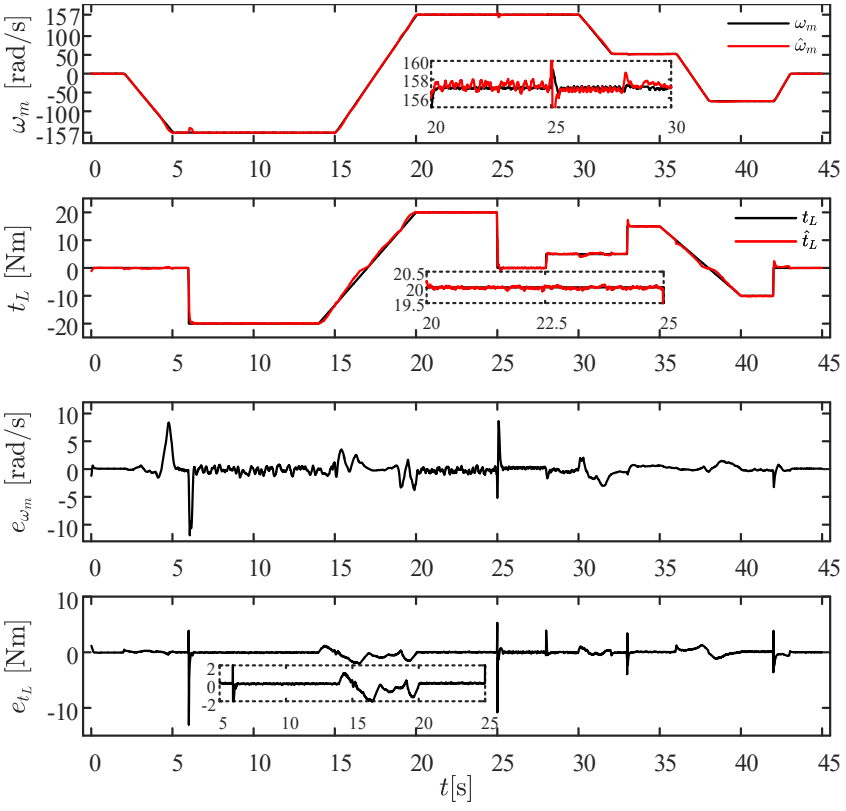

Fig. 8. Both-directions test dataset and estimation results for the proposed LSTM network. LSTM, long short-term memory.

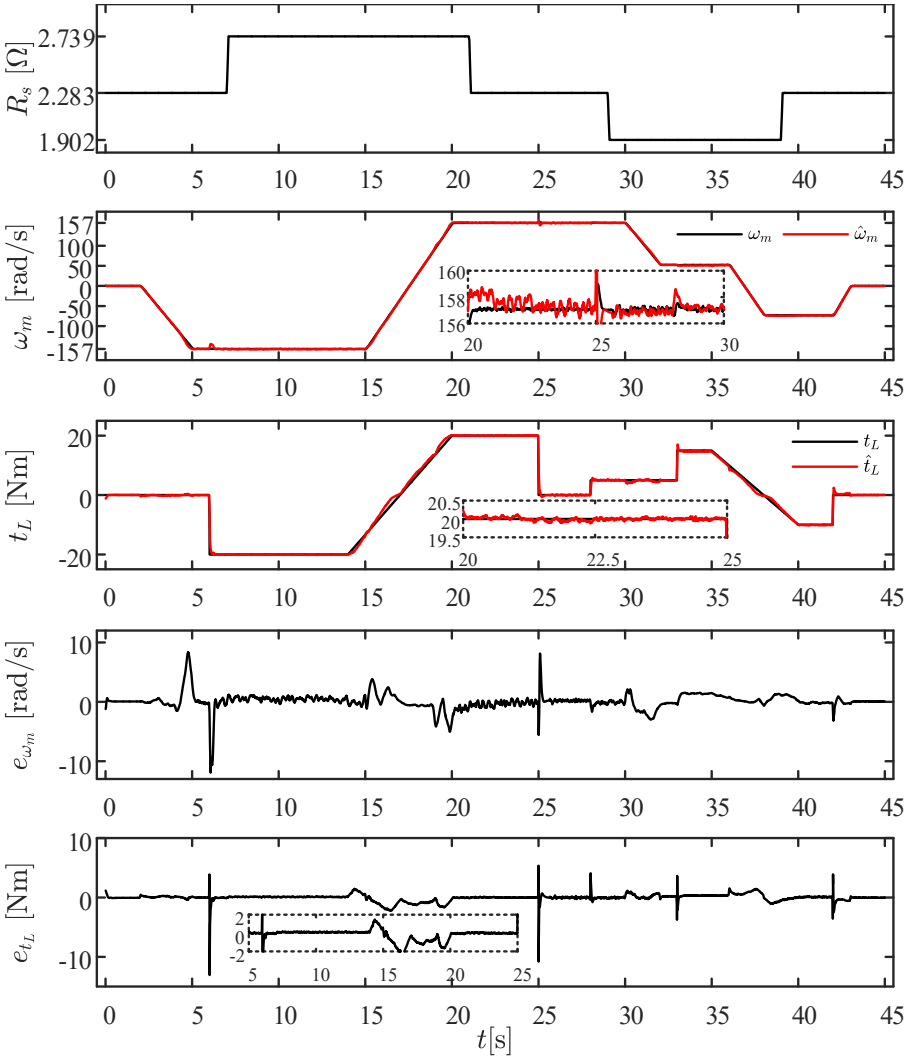

Fig. 9. Estimation results for the proposed LSTM network under the stator resistance changes. LSTM, long short-term memory.

promising ω_m and t_L estimation performance is observed for single-direction and both-directions tests in simulation studies, as indicated in Figures 7 and 8 as well as Table 5.

- Although linear and step-like t_L changes are performed for single-direction and both-directions tests, as indicated in Figures 7 and 8, the proposed LSTM network estimates the correct ω_m and t_L values, which can be confirmed based on the data presented in Table 5.
- In order to perform further examinations, the estimation performance of the proposed network is tested in the both-directions scenario with R_s and R_r variations. It is clear from Figures 9 and 10 as well as Table 5 that parameter variations have an adverse effect on the estimation performance of the proposed network.

In addition to the insights specified for the simulation results, owing to the usage of graphics processing unit (GPU) based hardware, it is possible for EV applications to use DNN-based solutions on the states and parameter estimation problem. Furthermore, considering the computational load required for the self-driving process including processing of light detection and ranging (LiDAR) sensor, camera, and radar data, it can be emphasised that the computational burden of the proposed method is not an additional problem for the GPU-based hardware used in the EV applications.

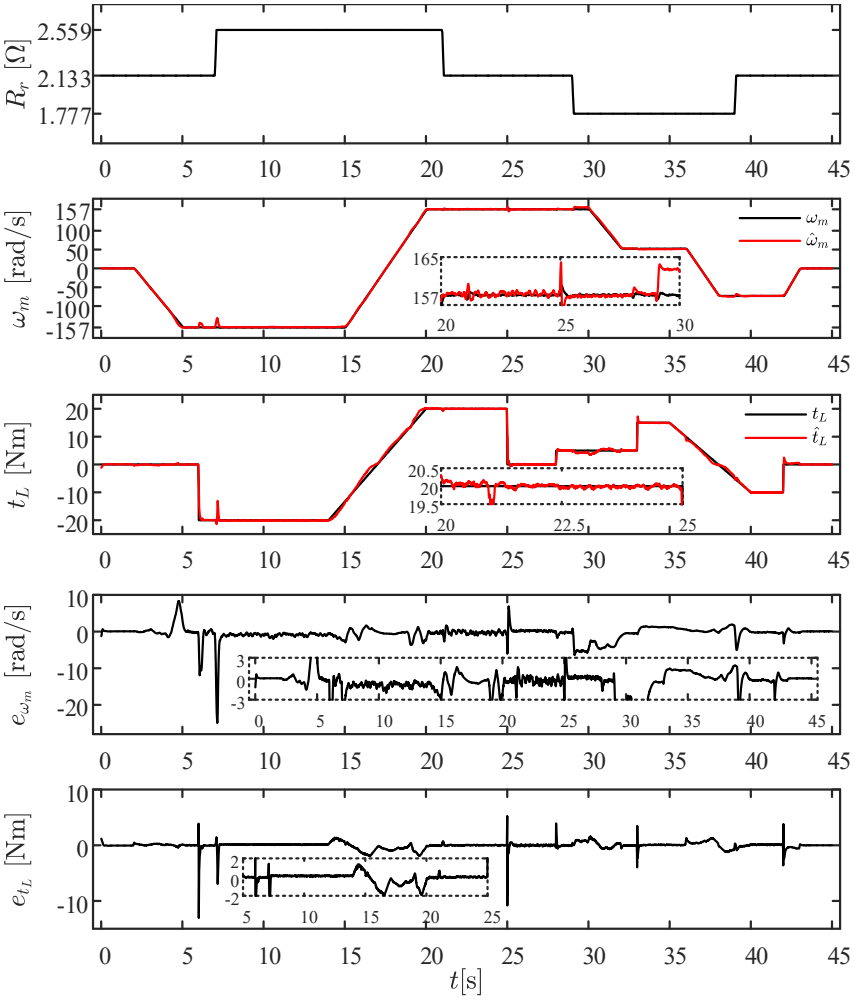

Fig. 10. Estimation results for the proposed LSTM network under the rotor resistance changes. LSTM, long short-term memory.

Figure 6	ω_m		t_L	
	RMSE	MAE	RMSE	MAE
w/o fine-tuning	1.307	0.699	0.514	0.285
With fine-tuning	1.282	0.646	0.499	0.261

MAE, mean absolute error; RMSE, root mean square error.

Table 4. The estimation results of the proposed method with and without fine tuning in single-direction and both-directions datasets.

	ω_m		t_L	
	RMSE	MAE	RMSE	MAE
Figure 7	1.340	0.882	0.478	0.296
Figure 8	1.282	0.645	0.499	0.261
Figure 9	1.370	0.745	0.557	0.316
Figure 10	2.112	1.054	0.570	0.309

MAE, mean absolute error; RMSE, root mean square error.

Table 5. The estimation performance of the proposed network is tested in the both-directions scenario with R_s and R_r variations.

5. Conclusion

In this paper, an LSTM-based DNN estimator using v_{sd}, v_{sq}, i_{sd}, and i_{sq} is designed and performed to estimate ω_m and t_L values of the IM for EV applications. To realise the training process, two different comprehensive scenarios are generated, and a fine-tuning process is also applied to increase the estimation performance of the proposed network. Furthermore, the different values of layer size and hidden size parameters of the network are examined to obtain the optimum parameters of the LSTM proposed in this paper. After the training and optimisation process of the proposed LSTM network was carried out, two different simulation test datasets, including operation of the IM in a single direction and both directions, are used to test the proposed network under the linear and step-like t_L changes. In order to show the estimation performance of the proposed network under IM parameter variations, the proposed LSTM network is tested under the R_s and R_r changes. The presented estimation results for both ω_m and t_L demonstrate that the performance of the proposed LSTM network against these challenging scenarios is highly promising. However, the estimation performance of the LSTM network is affected by the R_s and R_r variations. Thus, the robustness of the proposed LSTM network against the parameter variations needs to be increased in future research.

References

Acikgoz, H. and Korkmaz, D. (2021). Long short-term memory network-based speed estimation model of an asynchronous motor. In: *Proceedings of the 12th International Symposium on Advanced Topics in Electrical Engineering (ATEE)*, Bucharest, Romania, pp. 1–6.

Alsofyani, I. M. and Idris, N. R. N. (2016). Simple Flux Regulation for Improving State Estimation at Very Low and Zero Speed of a Speed Sensorless Direct Torque Control of an Induction Motor. *IEEE Transactions on Power Electronics*, 31(4), pp. 3027–3035.

Bednarz, S. A. and Dybkowski, M. (2019). Estimation of the Induction Motor Stator and Rotor Resistance using Active and Reactive Power Based Model Reference Adaptive System Estimator. *Applied Sciences*, 9(23), p. 5145.

Demir, R. (2023). Robust Stator Flux and Load Torque Estimations for Induction Motor Drives with EKF-Based Observer. *Electrical Engineering*, 105, pp. 551–562.

El Merrassi, W., Abounada, A. and Ramzi, M. (2021). Advanced Speed Sensorless Control Strategy for Induction Machine Based on Neuro-MRAS Observer. *Materials Today: Proceedings*, 45, pp. 7615–7621.

Ilten, E., Calgan, H. and Demirtas, M. (2022). Design of Induction Motor Speed Observer Based on Long Short-Term Memory. *Neural Computing and Applications*, 34(21), pp. 18703–18723.

Imane, G., Youcef, M., Abdelmadjid, G. and Zakaria, C. (2017). Neural Adaptive Kalman Filter for Sensorless Vector Control of Induction Motor. *International Journal of Power Electronics and Drive Systems (IJPEDS)*, 8(4), pp. 1841–1851.

Karanayil, B., Rahman, M. F. and Grantham, C. (2005). Stator and Rotor Resistance Observers for Induction Motor Drive using Fuzzy Logic and Artificial Neural Networks. *IEEE Transactions on Energy Conversion*, 20(4), pp. 771–780.

Karanayil, B., Rahman, M. F. and Grantham, C. (2007). Online Stator and Rotor Resistance Estimation Scheme using Artificial Neural Networks for Vector Controlled Speed Sensorless Induction Motor Drive. *IEEE Transactions on Industrial Electronics*, 54(1), pp. 167–176.

Kim, S. H., Park, T. S., Yoo, J. Y. and Park, G. T. (2001). Speed-Sensorless Vector Control of an Induction Motor Using Neural Network Speed Estimation. *IEEE Transactions on Industrial Electronics*, 48(3), pp. 609–614.

Orlowska-Kowalska, T., Dybkowski, M. and Szabat, K. (2010). Adaptive Sliding-Mode Neuro-Fuzzy Control of the Two-Mass Induction Motor Drive without Mechanical Sensors. *IEEE Transactions on Industrial Electronics*, 57(2), pp. 553–564.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J. and Chintala, S. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. *Advances*

in Neural Information Processing Systems, 32, pp. 8024–8035.

Reddy, B., Poddar, G. and Muni, B. P. (2022). Parameter Estimation and Online Adaptation of Rotor Time Constant for Induction Motor Drive. *IEEE Transactions on Industry Applications*, 58(2), pp. 1416–1428.

Rodriguez, J., Kennel, R. M., Espinoza, J. R., Trincado, M., Silva, C. A. and Rojas, C. A. (2012). High-Performance Control Strategies for Electrical Drives: An Experimental Assessment. *IEEE Transactions on Industrial Electronics*, 59(2), pp. 812–820.

Selvin, S., Vinayakumar, R., Gopalakrishnan, E. A., Menon, V. K. and Soman, K. P. (2017). Stock price prediction using LSTM, RNN and CNN-sliding window model. In: *Proceeding of the 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Udupi, India, pp. 1643–1647.

Simoes, M. G. and Bose, B. K. (1995). Neural Network Based Estimation of Feedback Signals for a Vector Controlled Induction Motor Drive. *IEEE Transactions on Industry Applications*, 31(3), pp. 620–629.

Sun, X., Chen, L., Yang, Z. and Zhu, H. (2013). Speed-Sensorless Vector Control of a Bearingless Induction Motor with Artificial Neural Network Inverse Speed Observer. *IEEE/ASME Transactions on Mechatronics*, 18(4), pp. 1357–1366.

Sundermeyer, M., Ney, H. and Schlüter, R. (2015). From Feedforward to Recurrent LSTM Neural Networks for Language Modeling. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(3), pp. 517–529.

Teler, K. and Orłowska-Kowalska, T. (2023). Analysis of the Stator Current Prediction Capabilities in Induction Motor Drive using the LSTM Network. *Power Electronics and Drives*, 8(1), pp. 31–52.

Ullah, A., Ahmad, J., Muhammad, K., Sajjad, M. and Baik, S. W. (2018). Action Recognition in Video Sequences using Deep Bi-Directional LSTM with CNN Features. *IEEE Access*, 6, pp. 1155–1166.

Vas, P. (1998). *Sensorless Vector and Direct Torque Control*. Oxford University Press, New York.

Verma, S., Henwood, N., Castella, M., Malrait, F. and Pesquet, J. C. (2020a). Modeling electrical motor dynamics using encoder-decoder with recurrent skip connection. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(02), British Columbia, Canada, pp. 1387–1394.

Verma, S., Henwood, N., Castella, M., Jebai, A. K. and Pesquet, J. C. (2020b). Neural networks based speed-torque estimators for induction motors and performance metrics. In: *Proceedings of the 46th Annual Conference of the IEEE Industrial Electronics Society*, Singapore, pp. 495–500.

Verma, S., Henwood, N., Castella, M., Jebai, A. K. and Pesquet, J. C. (2023). Neural Speed–Torque Estimator for Induction Motors in the Presence of Measurement Noise. *IEEE Transactions on Industrial Electronics*, 70(1), pp. 167–177.

Vicente, I., Endeman, A., Garin, X. and Brown, M. (2010). Comparative Study of Stabilising Methods for Adaptive Speed Sensorless Full-Order Observers with Stator Resistance Estimation. *IET Control Theory Applications*, 4(6), pp. 993–1004.

Wróbel, K., Serkies, P. and Szabat, K. (2020). Model Predictive Base Direct Speed Control of Induction Motor Drive—Continuous and Finite Set Approaches. *Energies*, 13(5), p. 1193.

Yildiz, R., Barut, M. and Zerdali, E. (2020a). A Comprehensive Comparison of Extended and Unscented Kalman Filters for Speed-Sensorless Control Applications of Induction Motors. *IEEE Transactions on Industrial Informatics*, 16(10), pp. 6423–6432.

Yildiz, R., Barut, M. and Demir, R. (2020b). Extended Kalman Filter Based Estimations for Improving Speed□Sensored Control Performance of Induction Motors. *IET Electric Power Applications*, 14(12), pp. 2471–2479.

Yildiz, R., Demir, R. and Barut, M. (2023). Online Estimations for Electrical and Mechanical Parameters of the Induction Motor by Extended Kalman Filter. *Transactions of the Institute of Measurement and Control*, Early access, Available at: https://doi.org/10.1177/01423312231160582.

Yin, Z., Bai, C., Du, N., Du, C. and Liu, J. (2021). Research on Internal Model Control of Induction Motors Based on Luenberger Disturbance Observer. *IEEE Transactions on Power Electronics*, 36(7), pp. 8155–8170.

Zhang, Y., Yin, Z., Zhang, Y., Liu, J. and Tong, X. (2020). A Novel Sliding Mode Observer with Optimized Constant Rate Reaching Law for Sensorless Control of Induction Motor. *IEEE Transactions on Industrial Electronics*, 67(7), pp. 5867–5878.