Neural speed–torque estimator for induction motors in the presence of measurement noise

Sagar Verma, Nicolas Henwood, Marc Castella, Al Kassem Jebai, Jean-Christophe Pesquet

Neural Speed-Torque Estimator for Induction Motors in the Presence of Measurement Noise

Sagar Verma, Nicolas Henwood, Marc Castella, *IEEE Member*,
Al Kassem Jebai, and Jean-Christophe Pesquet, *IEEE Fellow*

***Abstract*—In this paper, a neural network approach is introduced to estimate non-noisy speed and torque from noisy measured currents and voltages in induction motors with Variable Speed Drives. The proposed estimation method is comprised of a neural speed-torque estimator and a neural signal denoiser. A new training strategy is introduced that combines large amount of simulated data and a small amount of real world data. The proposed denoiser does not require non-noisy ground truth data for training, and instead uses classification labels which are easily generated from real-world data. This approach improves upon existing noise removal techniques by learning to denoise as well as classify noisy signals into static and dynamic parts. The proposed neural network based denoiser generates clean estimates of currents and voltages which are then used as inputs to the neural network estimator of speed and torque. Extensive experiments show that the proposed joint denoising-estimation strategy performs very well on real data benchmarks. The proposed denoising method is shown to outperform several widely used denoising methods and a proper ablation study of the proposed method is conducted.**

***Index Terms*—Denoiser, Induction Motors, Neural Networks, Speed-Torque Estimator, Variable Speed Drives**

I. Introduction

This paper considers induction motors controlled by Variable Speed Drives (VSD). The complex physics of induction motors requires a controller that can provide robust control based on its dynamics. Induction motor controllers also offer protection and supervision of the electro-mechanical system [1]. For these services, it is required to establish models based on the physical dynamics of induction motors. These dynamical models are dependent on different induction motor quantities like currents, voltages, speed, torque, fluxes, inductances or resistances, which are measured directly (sensors) or indirectly (estimators). Accurately measuring some of these quantities may be challenging due to temperature variations or the presence of noise for instance [2].

Sagar Verma and Jean-Christophe Pesquet are with Université Paris-Saclay, CentraleSupélec, Inria, Centre de Vision Numérique. Sagar Verma is also part of Schneider Toshiba Inverter Europe (e-mail: sagar.verma@centralesupelec.fr, sagar.verma@se.com; jean-christophe.pesquet@centralesupelec.fr).

Marc Castella is with SAMOVAR, Télécom SudParis, Institut Polytechnique de Paris (e-mail: marc.castella@telecom-sudparis.eu).

Nicolas Henwood and Al Kassem Jebai are with Schneider Toshiba Inverter Europe (e-mail: nicolas.henwood@se.com; alkassem.jebai@se.com).

Modeling of electrical motors based on analytical mechanics and energy consumption is presented in [3]. In particular, existing methods for designing a controller for an induction motor depend on whether a perfect knowledge of the parameters is available [4] or if associated uncertainties are considered [5]. Some methods take statistical model of noisy measurements into account along with the analytical model, when used for controller [6], [7] or fault detection [8], [9].

Neural network-based control methods [10], [11] have also been proposed to learn models that can alleviate problems associated with the traditional model-based approach. In [12], an encoder-decoder network is proposed to learn dynamics of electrical motors directly from recorded data. The proposed method achieves good performance in modeling the input-output relationship. In [13] a proper study and benchmarking of different networks proposed in [12] has been performed using both machine learning and electrical engineering performance metrics. These studies do not take however into account the presence of measurement noise which is always present in acquisition processes.

Noise reduction in time signals is a very evolved field with a multitude of research involving various methodologies. Some of the techniques that can be easily applied are linear smoothing filters and non-linear filters [14]. Kalman filter [15] is widely used in noisy observation where a state-space based estimation is done which takes the system model as input. Also, there are transform based methods like those based on wavelet transforms [16], [17] which remove noisy components from transformed sensor data. Variational methods often based on the total variation [18]–[20] have also been used in signal denoising and change detection, providing a robust and often more flexible solution over linear filter based denoisers. Kalman filter, transform-based and variational methods require prior knowledge about the noise/signal statistics for efficient denoising. Deep learning based method like stacked autoencoders [21] have been shown to be promising in learning to remove noise. However, neural network denoisers require a large amount of data with non-noisy ground truth, which is not possible in the case of induction motors.

The main contributions of this paper are the following:

- A speed-torque estimation pipeline for supervision and monitoring tasks in VSDs is proposed that

processes noisy currents-voltages and outputs non-noisy currents-voltages, which are then used to predict non-noisy speed-torque.
- A novel denoising method called Meta-Denoiser is proposed which overcomes limitations of standard denoisers, and provides a robust solution to measurement noise removal from currents and voltages.
- It is shown that a large amount of simulated data and a limited amount of real-world data can be used to train our pipeline and achieve good performance.

The paper is structured as follows: Section II describes neural network based speed-torque estimator and its evaluation metrics. Section III presents the test bench and the collected dataset. Denoising methods are introduced in Section IV. Experiments are reported and analysed in Section V, followed by conclusions in Section VI.

II. Speed-Torque Estimator

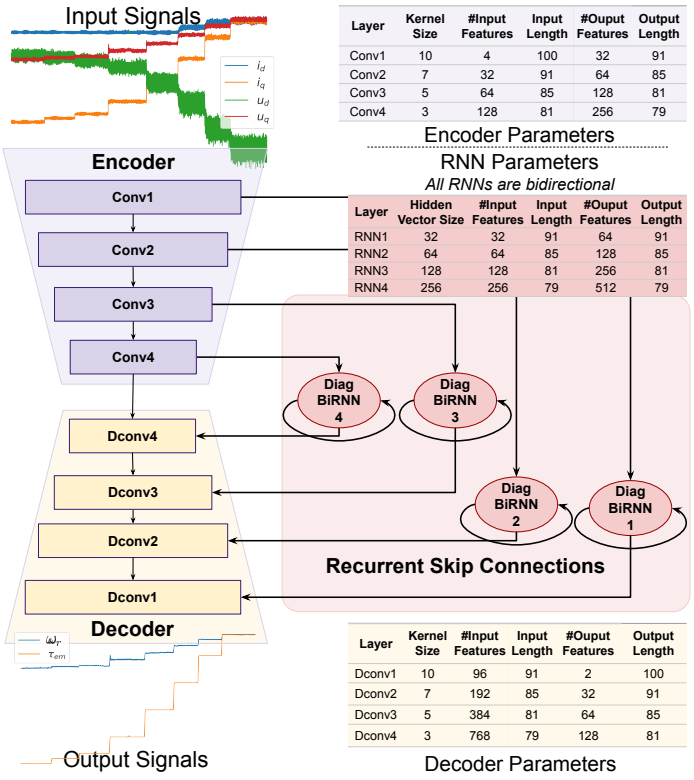
Learning input-output relationships using neural networks has been explored in [12]. Several benchmark methods that are derivatives of standard neural networks have been investigated. Broadly, feed-forward network, convolutional neural network, recurrent neural network (RNN) and Long-Short Term Memory (LSTM) structures have been evaluated. In [13], these networks have been analysed using various metrics in ideal conditions where noise is absent. The focus of this work is on deriving rotor speed ω_r and electromagnetic torque τ_{em} from currents (i_d,i_q) and voltages (u_d,u_q) in the rotating dq frame, from a real motor. When using a VSD, the motor voltages are applied to the motor and the motor currents are measured using integrated current sensors. However, the important physical quantities for control are the speed and the torque. Thus, getting these unknown quantities from the applied/measured values of the voltages and currents is of primary interest. This paper is an extension of the work done in [13] by taking into account measurement noise in currents and voltages. Encoder-decoder with Bidirectional Diagonalized Recurrent Skip Connection (DiagBiRNN) network from [12], [13] has been used for all our experiments.

DiagBiRNN, shown in Figure 1, is an encoder-decoder network with 1D convolutions in the encoder and 1D deconvolutions in the decoder. Encoder-decoder layers are connected via skip connections, consisting of recurrent units which further help the network to learn temporal information in latent space. The hidden state update equation of an RNN is given by

$$h_t = \tanh(Wx_t + Uh_{t-1} + b), \tag{1}$$

where $x_t \in \mathbb{R}^M$ and $h_t \in \mathbb{R}^N$ are the input and hidden states at time t, M and N the respective sizes of the RNN input and output vectors. $W \in \mathbb{R}^{N\times M}$, $U \in \mathbb{R}^{N\times N}$ and $b \in \mathbb{R}^N$ are weights and bias.

One problem with handling multiple recurrent neural units is the large number of parameters involved in the matrix multiplications between weights and features. DiagBiRNN overcomes this limitation by not using vanilla RNNs but diagonalizing weights in the recurrent units: W and U are replaced by vectors and all matrix multiplication operations are replaced by Hadamard products.



Layer	Kernel Size	#Input Features	Input Length	#Ouput Features	Output Length
Conv1	10	4	100	32	91
Conv2	7	32	91	64	85
Conv3	5	64	85	128	81
Conv4	3	128	81	256	79

Layer	Hidden Vector Size	#Input Features	Input Length	#Ouput Features	Output Length
RNN1	32	32	91	64	91
RNN2	64	64	85	128	85
RNN3	128	128	81	256	81
RNN4	256	256	79	512	79

Layer	Kernel Size	#Input Features	Input Length	#Ouput Features	Output Length
Dconv1	10	96	91	2	100
Dconv2	7	192	85	32	91
Dconv3	5	384	81	64	85
Dconv4	3	768	79	128	81

Fig. 1: DiagBiRNN based speed-torque estimator. All the experiments in this paper use this network to predict speed and torque from currents and voltages.

A. Machine Learning Metrics

To analyse model performance at global scope, some standard machine learning (ML) metrics have been used, namely the mean absolute error (MAE) and the symmetric mean absolute percentage error (SMAPE) whose expressions are recalled below:

$$\text{MAE}(y,\hat{y}) = \frac{1}{T}\sum_{t=1}^{T}|y_t - \hat{y}_t|, \tag{2}$$

$$\text{SMAPE}(y,\hat{y}) = \frac{100}{T}\sum_{t=1}^{T}\frac{|\hat{y}_t - y_t|}{|\hat{y}_t| + |y_t|}, \tag{3}$$

where y_t is the ground truth, $\hat{y}_t$ is the predicted output of the model at time t, and T is the total experiment duration.

B. Electrical Engineering Performance Metrics

A more insightful way of evaluating the performance from an industrial standpoint is to compute widely used electrical engineering metrics. Following metrics have been used for the response signal to a speed or

torque reference ramp (whose amplitude is the absolute difference between the starting and target values):

- **2% response time ($t_{2\%}$)** is the time value at which the response signal has covered 2% of the ramp amplitude.
- **95% response time ($t_{95\%}$)** is the time value after which the response signal remains at less than 5% of the ramp amplitude from the target value.
- **Overshoot ($D\%$)** is the difference between the maximum peak value of the response signal and the final steady-state value. It is expressed in percentage of the ramp amplitude.
- **Steady-state error (E_{ss})** is the difference between the response signal and target values once the steady-state has been reached.
- **Following error (E_{fol})** is the difference between the reference and response signal values when the reference value has covered 50% of the ramp amplitude.
- **Maximum acceleration torque (speed ramp) ($\Delta\tau_{max}$)** is the max response torque deviation during the speed ramp.
- **Speed drop (torque ramp) (SD)** is the max response speed deviation during the torque ramp.

III. Dataset

A. Experimental Workbench

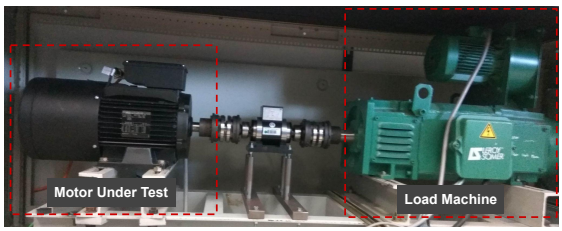


Fig. 2: Experimental setup

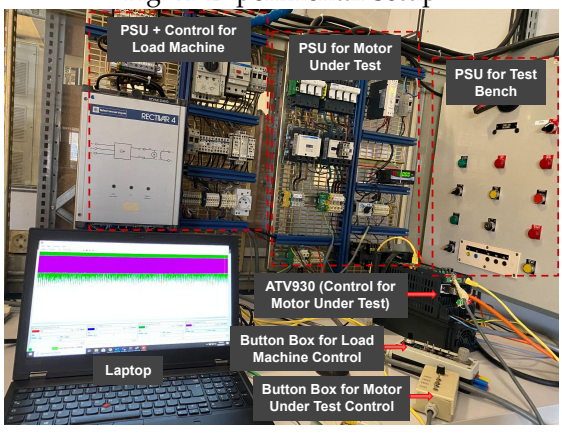


Fig. 3: Test bench setup containing the data acquisition

Figure 2 shows a 4kW induction motor under test, and a direct current motor as load machine part of our experimental setup. Figure 3 shows our test bench setup which consists of the power supply units (PSU) for the test bench, the motor under test, the oad machine, and an ATV930 VSD to control the induction motor. In addition, button boxes allow us to give run orders and manually set speed / torque references, both for the motor under test and the load machine.

B. Collected Dataset

This paper uses the reference trajectory generator proposed in [13] to generate 100 simulated trajectories of 150 minutes for the training set and 50 simulations of 30 minutes for the validation set. Training and validation sets are well separated to avoid over-fitting. Simulink model of a 4kW induction motor is then used to simulate these trajectories to collect data at 4ms intervals. The simulation dataset consists of the following electrical quantities: currents i_d and i_q, voltages u_d and u_q, rotor speed ω_r, and electromagnetic torque τ_{em}.

In addition to these simulated data, 10 experiments were performed on our 1.5kW and 4kW motors (50Hz nominal speed) shown in Figure 2 to collect real data. These experiments have different types of trajectories: constant speed with torque variations in [-120, 120]% of the nominal torque, speed variations in [-70, 70] Hz at no load, torque steps under constant speed and, in some cases, both speed and torque vary. All these types of trajectories have been considered to cover the majority of the use cases that may arise in real world. This provides different kinds of dynamic behaviour which makes denoising and speed-torque estimation quite challenging.

To understand the effect of noise on the performance of DiagBiRNN, experiments with simulated and real motor benchmarks have been performed. To evaluate DiagBiRNN in simulation and on real (prefix R) motor data, following benchmarks have been used:

(a) **Dynamic-Speed1 / RDynamic-Speed1:** Reference speed goes from 0 to 50Hz in 1 second at no load.
(b) **Dynamic-Speed2:** Reference speed goes from 0 to 50Hz in 0.5 second under nominal torque.
(c) **Quasi-Static:** Two quasi-static cases are considered: **1)** at no load and **2)** under constant 50% of nominal torque. Reference speed goes from 70 to -70 Hz in 50 seconds in both cases.
(d) **RDynamic-Speed2**: Reference speed goes through multiple inversions in both direction.
(e) **RDynamic-Torque:** Load torque goes from 0 to 100% of nominal torque in one time step at a constant 50Hz reference speed.
(f) **RQuasi-Static:** At no load, reference speed goes from 80 to -80Hz in 50 seconds.

IV. Denoising Methods

This section first presents the measurement noise modeling process. Then various strategies for denoising currents and voltages are discussed along with the proposed method.

A. Noise Modeling

The noise of currents (i_d,i_q) and voltages (u_d, u_q) have been modeled in a way similar to [22]. To that extent,

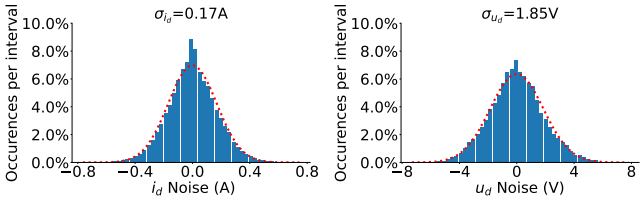


Fig. 4: Noise distributions of i_d and u_d from real data

static parts of the experimental data have been collected, i.e. the periods during which both speed and torque are constant. For each static part, assuming that the non-noisy "true" currents and voltages signals coincide with their mean values on the static part, the temporal noise signals for the two currents and the two voltages have been determined. Gathering all the static parts of our experimental data, the temporal distributions of the noise corrupting each signal have been obtained. Figure 4 shows the distributions of noise in current i_d and voltage u_d. Relying on ergodicity and making the assumption that the real distributions can be approximated by Gaussian ones, the statistical characteristics of the noises have been deduced. The Gaussian approximations are shown by the red lines in Figure 4. This shows that the noises approximately follow normal distributions with zero mean and standard deviations equal to $\sigma_{i_d} = 0.17A$, $\sigma_{i_q} = 0.29A$, $\sigma_{u_d} = 1.85V$, and $\sigma_{u_q} = 1.78V$, respectively.

B. Standard Denoisers

Extended Kalman filter (EKF) [23] is a state-space based non-linear filtering approach. A diagonal measurement noise co-variance matrix is chosen using the noise variances estimated in Section IV-A and the state transition matrix is set to $[[1,0],[0,1]]$. Wavelet transform (WT) denoising is a non-linear estimation method operating on each wavelet coefficient separately. The adaptive Bayes Shrink algorithm [24] has been used to soft threshold wavelet coefficients using the noise standard deviations identified in Section IV-A. Minmax-concave total variation (MCTV) [20] is a non-linear variational method. It has been used with $K = 100$ maximum iterations, a root mean square error of 10^{-3} as convergence criteria, a regularization constant $\lambda = \sqrt{\sigma T}/5$ with σ the standard deviations of Section IV-A and T the duration of the signal, and the non-convexity parameter $\alpha_{nc} = 0.3/\lambda$, after having also tried other values for the numerator.

Unlike EKF, WT and MCTV, denoising auto-encoder (DAE) is a neural network based technique. It consists of 3 layers of 1-D convolutions and 3 layers of 1-D deconvolutions with a number of channels equal to 1-32-64-128-64-32-1.

Although our experiments in the next section will show that these four standard denoisers work quite well on simulated data, this is not the case on real motor data. EKF often fails when there is a sudden transition in noise amplitude which could be resolved using Adaptive EKF [25]. The denoised outputs of WT and MCTV exhibit staircase effects in ramp parts of the signal (with less error for MCTV). DAE solves these problems and gives the smoothest denoised output, but leads to incorrect predictions at the start and the end of a ramp.

C. Proposed Meta-Denoiser (MD)

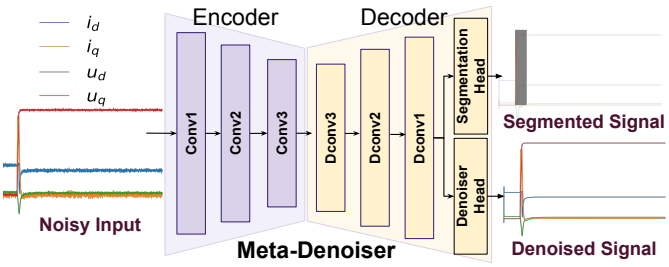


Fig. 5: Meta-Denoiser architecture.

To overcome the problems of measurement noise this paper introduces Meta-Denoiser (MD) shown in Figure 5. MD is an extension of DAE where the last deconvolution layer gives the same number of feature channels as its input. This is then fed to the segmentation head and the denoiser head to predict segmentation and denoised outputs. By training it to identify dynamic and static parts, the model is able to identify the correct start and end of a ramp, thereby overcoming the problem of DAE.

MD is trained using the MD-Joint loss function:

$$\mathcal{L}_{\text{MD-Joint}} = \alpha \mathcal{L}_{\text{BCE}} + (1-\alpha)\mathcal{L}_{\text{MSE}} \quad \begin{cases} \alpha = 0.5 & \text{if } y^i \in \text{Sim} \\ \alpha = 1 & \text{if } y^i \in \text{Real} \end{cases} \tag{4}$$

$$\mathcal{L}_{\text{MSE}} = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{1}{T}\sum_{t=1}^{T}(y_t^i - \hat{y}_t^i)^2\right) \tag{5}$$

$$\mathcal{L}_{\text{BCE}} = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{1}{T}\sum_{t=1}^{T}(-z_t^i \log(\hat{z}_t^i) - (1-z_t^i)\log(1-\hat{z}_t^i))\right) \tag{6}$$

where y_t^i and $\hat{y}_t^i$ are the respective values of output and predicted sample i at time-step t for the denoising task, z_t^i and $\hat{z}_t^i$ are the respective classification probabilities of output and predicted sample i at time-step t for the segmentation task, and N is the number of training samples where each sample is of duration T. Eq. (4) is made of a MSE part expressed in Eq. (5), accounting for the denoising (regression) task, and a widely used binary cross entropy (BCE) loss [26] in Eq. (6).

V. Experiments and Results

All the experiments in this paper have been performed on an Ubuntu 18.04 OS with Nvidia V100 GPU. To train MD and DiagBiRNN, the dataset has been partitioned into four parts: 70% (resp. 30%) of the simulation data are used for training (resp. validation), whereas 20% (resp. 80%) of the real motor data for fine-tuning (resp. testing) purpose. The fact that the majority of the real data has

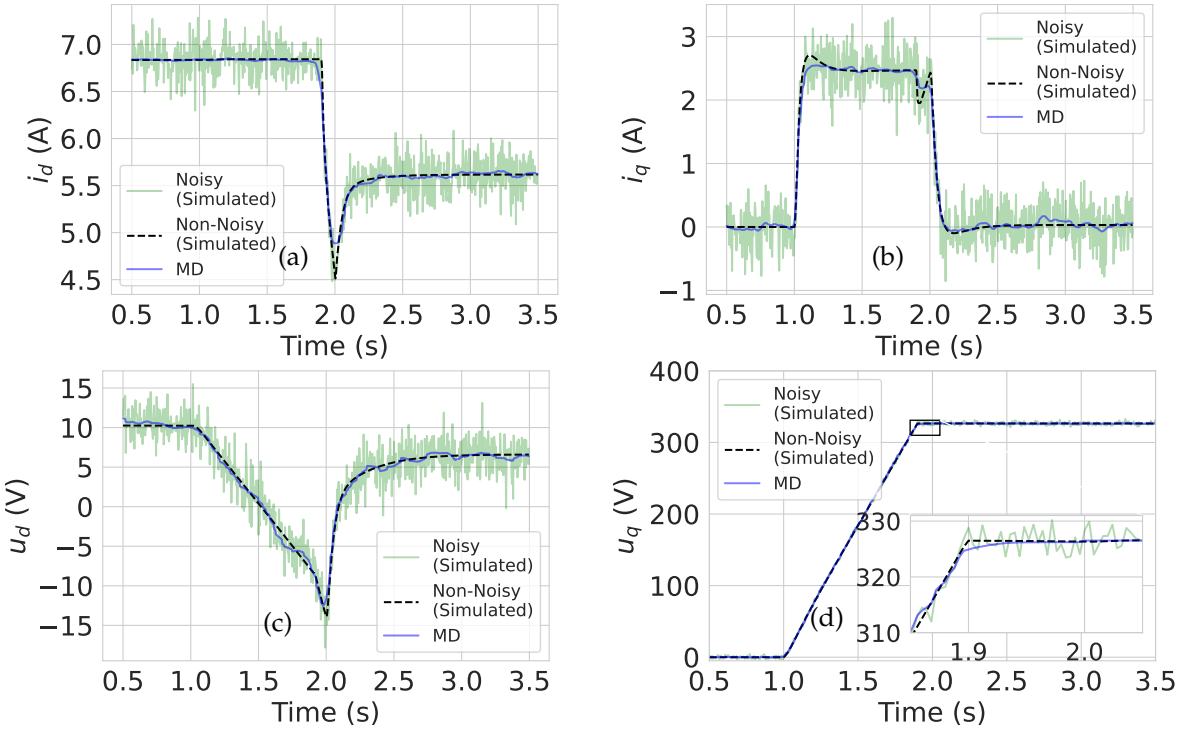


Fig. 6: Meta-Denoiser (MD) performance on Dynamic-Speed1 simulated benchmark.

been used for testing is in line with real industrial needs. DAE and MD have been trained with mean square error (MSE) loss function and a stochastic gradient descent optimizer with the following hyperparameters: 100 epochs, learning rate of 0.01, and batch size of 128. DiagBiRNN uses training strategy and hyperparameters described in [12], [13]. EKF, WT, MCTV, DAE and MD have been used to denoise the currents (i_d, i_q) and voltages (u_d, u_q). The denoised currents and voltages are then used to predict the speed ω_r and the torque τ_{em} using DiagBiRNN trained on non-noisy simulated data [13]. For online inference both networks were implemented using [27].

Method	Speed ω_r		Torque τ_{em}	
	MAE	SMAPE	MAE	SMAPE
CASE A	0.03	18.7%	0.04	38.5%
CASE B	0.05	20.1%	0.13	41.3%
CASE C	0.05	19.7%	0.13	41.0%
EKF [23]	0.05	19.4%	0.13	41.0%
WT [16]	0.05	19.4%	0.12	41.0%
MCTV [20]	0.05	19.4%	0.12	40.1%
DAE [21]	0.04	19.0%	0.09	39.7%
MD (Ours)	**0.04**	**18.8%**	**0.05**	**38.9%**

TABLE I: Aggregated ML metrics for the predictions done on all simulated benchmarks

A. Simulated Benchmarks

To analyze the effect of noise on DiagBiRNN, a set of experiments with simulated data in both absence and presence of noise and with real motor data have been performed. Based on different combinations of test data and training conditions of DiagBiRNN, the following cases are studied:

1) **Case A:** NN estimator trained on non-noisy simulated data, applied on non noisy simulated data.
2) **Case B:** Estimator from Case A applied to noisy simulated data.
3) **Case C:** Estimator trained on simulated data with noisy currents and voltages and non-noisy speed and torque, applied on noisy simulated data.
4) **Case D:** Transfer learning presented in [12] is used to train and predict on real motor data.

Table I shows the aggregated ML metrics obtained by DiagBiRNN on the simulated benchmarks with and without denoising using various methods. Small MAE and SMAPE values are desirable for a good prediction model. It can be observed that Case B performs worse than Case A since DiagBiRNN alone has no way of cancelling the noise. Case C performs relatively better than Case B since training is performed on noisy simulated data. Amongst standard denoiser, minor improvements with MCTV and DAE are observed. However, the proposed MD offers the best performance among all the evaluated methods, with quantitative results approaching the ones obtained in the non-noisy Case A.

Table II shows performance metrics for Dynamic-Speed1 benchmark. Our objective is to be as close as possible to "Real" values given in the first row, "Real" standing here for "non-noisy (simulated)". It can be seen that MD-DiagBiRNN is the closest one to the "Real" for almost all metrics, again leading in estimating speed and torque.

Figure 6 shows results of denoising for Dynamic-Speed1 benchmark. Sub-figures (a), (b), (c), and (d) dis-

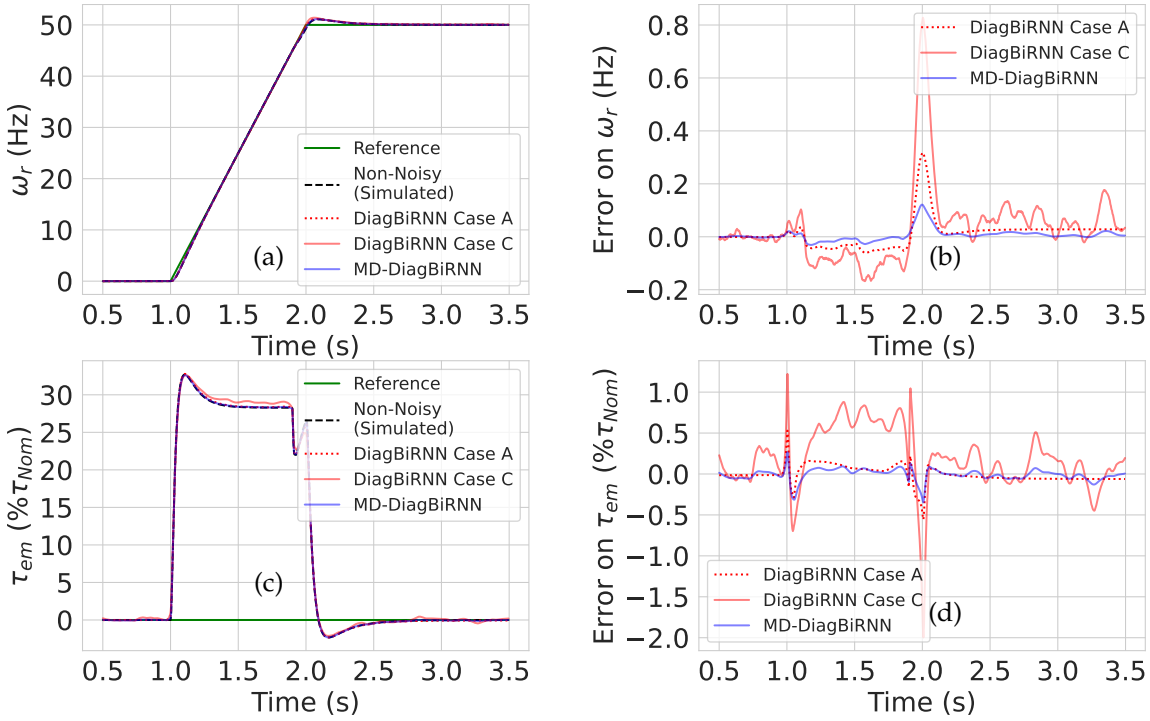


Fig. 7: Speed-torque estimator (DiagBiRNN) performance on Dynamic-Speed1 simulated benchmark applied on non-noisy (Case A) or noisy currents and voltages (without - Case C - or with our proposed meta-denoiser - MD)

Method	$t_{2\%}$ (ms)	$t_{95\%}$ (ms)	E_{fol} (Hz)	$D\%$ (%)	E_{ss} (Hz)	$\Delta\tau_{max}$ (%τ_{nom})
"Real"	**44.0**	**960**	**-0.02**	**2.16**	**0.00**	**32.7**
CASE A	44.1	956	0.01	2.32	0.03	32.8
CASE B	44.2	**950**	**-0.01**	2.26	**-0.01**	32.8
CASE C	44.2	955	**0.03**	**2.39**	**0.04**	32.8
EKF [23]	**44.1**	952	**0.02**	**2.48**	**0.04**	32.6
WT [16]	44.2	955	**0.03**	**2.37**	**0.04**	**32.7**
MCTV [20]	**44.1**	956	0.02	2.34	**0.04**	**32.7**
DAE [21]	43.8	**950**	0.00	2.33	0.02	**32.7**
MD (Ours)	44.2	**958**	**-0.01**	**2.23**	**0.01**	**32.7**

TABLE II: Performance metrics for the predictions performed on simulated Dynamic-Speed1 benchmark

Method	$t_{2\%}$ (ms)	$t_{95\%}$ (ms)	E_{fol} (Hz)	$D\%$ (%)	E_{ss} (Hz)	$\Delta\tau_{max}$ (%τ_{nom})
"Real"	**32.0**	**492**	**0.32**	**3.86**	**0.00**	**65.7**
CASE A	**32.1**	**588**	0.24	5.77	**0.05**	**65.6**
CASE B	32.2	**596**	**0.30**	6.28	**0.18**	65.9
CASE C	32.2	**604**	**0.20**	**7.33**	0.11	**69.8**
EKF [23]	32.2	**616**	**0.19**	**7.23**	0.12	**65.8**
WT [16]	32.2	**604**	**0.20**	**7.27**	0.12	**65.8**
MCTV [20]	**32.1**	**600**	0.22	6.24	0.07	65.5
DAE [21]	**32.9**	**608**	**0.19**	6.21	0.10	66.3
MD (Ours)	**32.1**	**484**	0.26	**4.45**	**0.05**	65.9

TABLE III: Performance metrics for the predictions performed on simulated Dynamic-Speed2 benchmark

play non-noisy and noisy simulated currents and voltages as well as MD denoised signals. It can be observed that MD denoised trajectories are close to non-noisy simulated trajectories, demonstrating the good denoising performance of MD.

Figure 7 shows speed-torque estimation results. Sub-figures (a) and (c) show speed and torque prediction results in Case A (applied to non-noisy simulated), Case C (applied to noisy simulated), and MD-DiagBiRNN (applied to MD denoised currents and voltages). For better evaluation, sub-figures (b) and (d) show the difference between the generated outputs and non-noisy simulated. While Case C presents substantial errors, it can be seen that MD-DiagBiRNN has errors of the same order of magnitude as in non-noisy Case A.

Table III and Figure 8 show performance metrics and results of speed-torque estimation for Dynamic-Speed2 benchmark. Similar to Dynamic-Speed1 benchmark it

Method	CASE A	CASE B	CASE C
Quasi-Static1	0.198	0.213	0.227
Quasi-Static2	0.171	0.199	0.204

Method	EKF	WT	MCTV	DAE	MD (Ours)
Quasi-Static1	0.196	0.201	0.198	0.198	**0.197**
Quasi-Static2	0.184	0.189	0.181	0.174	**0.173**

TABLE IV: Maximum absolute error (Hz) for the predictions done on simulated Quasi-Static benchmarks

can be observed that MD-DiagBiRNN performs very well.

Table IV reports the maximum absolute error between the model predicted speed and the non-noisy simulated speed obtained with the different methods on two quasi-static benchmarks. The lowest error values are reached for non-noisy Case A and for DiagBiRNN associated with NN denoisers, namely DAE, and once again MD.

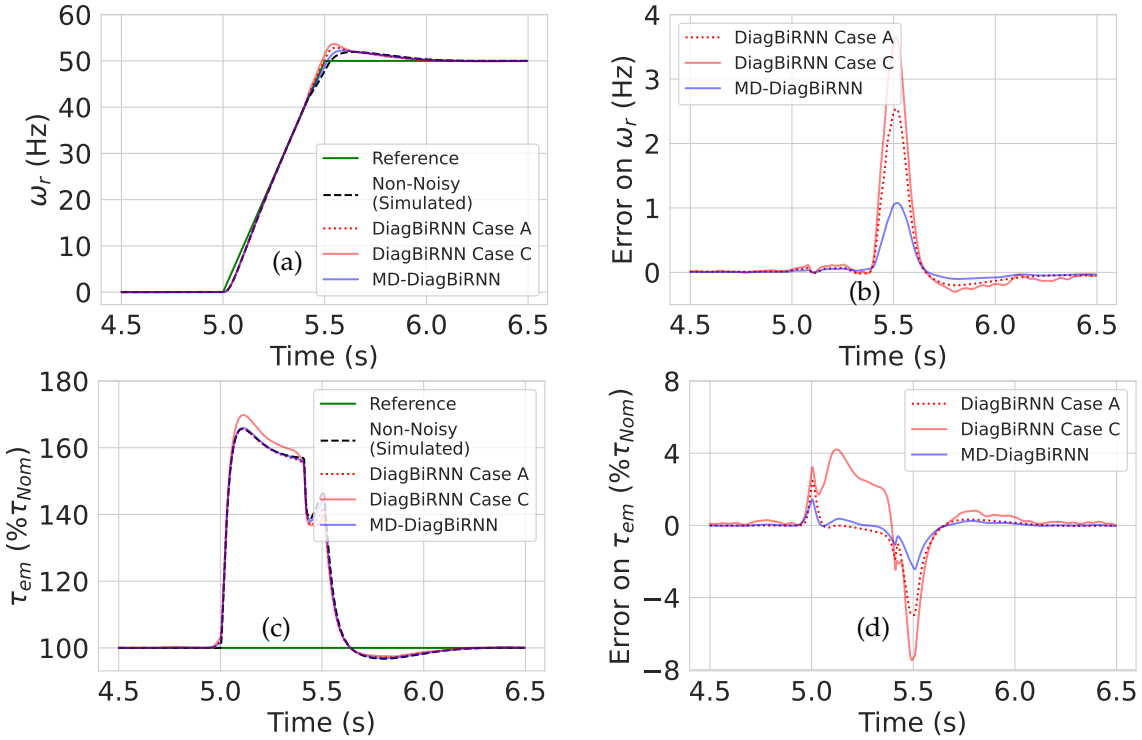


Fig. 8: Speed-torque estimator (DiagBiRNN) performance on Dynamic-Speed2 simulated benchmark applied on non-noisy (Case A) or noisy currents and voltages (without - Case C - or with our proposed meta-denoiser - MD)

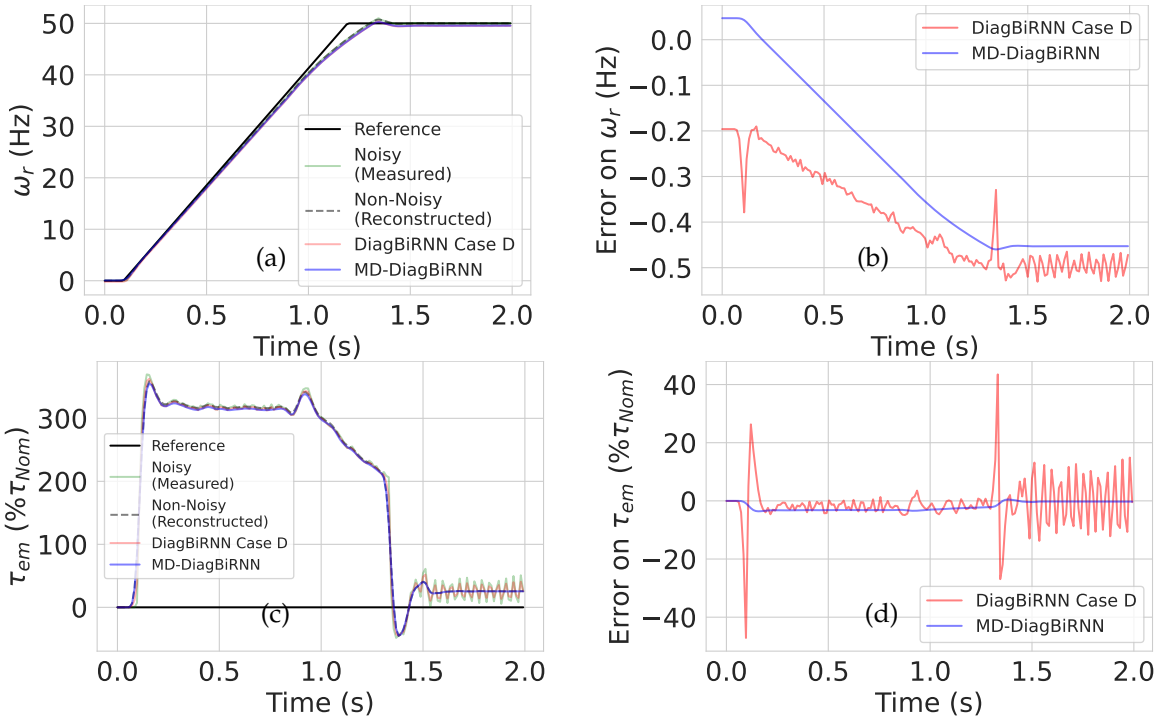


Fig. 9: Speed-torque estimator (DiagBiRNN) online performance on RDynamic-Speed1 real data benchmark (on 1.5kW motor), with transfer learning (Case D) or with our proposed meta-denoiser (MD).

B. Real Data Benchmarks

The main requirement for the speed-torque estimator is that it operates in real-time with minimum possible delay. The processing time for DiagBiRNN is 40ms. To avoid adding too much extra delay, it is recommended that denoisers operate in less than 40ms. This is possible in the case of EKF, DAE, and MD. WT and MCTV require larger delay to get acceptable performance, making them hardly usable for real-time usage. Since EKF and DAE do not perform as well as MD on simulated benchmarks, real benchmark experiments are limited to Case D and MD.

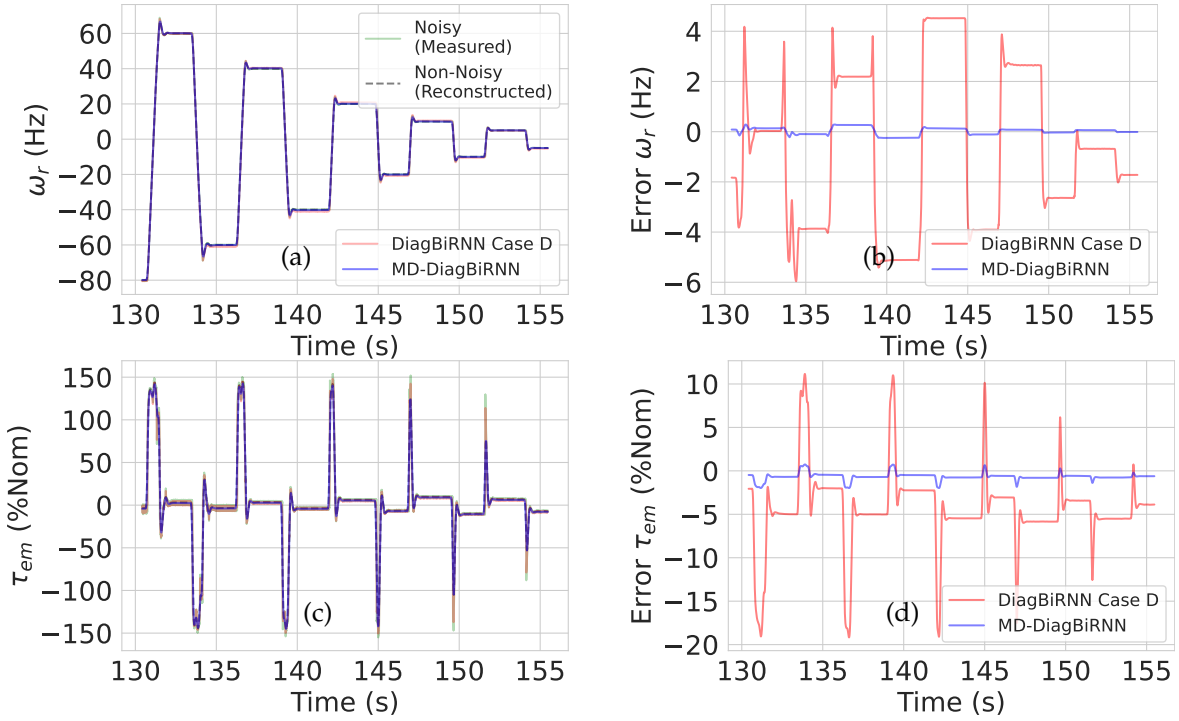


Fig. 10: Speed-torque estimator (DiagBiRNN) performance on RDynamic-Speed2 real data benchmark, with transfer learning (Case D) or with our proposed meta-denoiser (MD)

Method	Speed ω_r		Torque τ_{em}	
	MAE	SMAPE	MAE	SMAPE
CASE D	0.92	23.7%	1.12	39.2%
MD (Ours)	**0.61**	**13.7%**	**1.09**	**35.2%**

TABLE V: Aggregated ML metrics for the predictions done on all real data benchmarks

Method	$t_{2\%}$ (ms)	$t_{95\%}$ (ms)	E_{fol} (Hz)	$D\%$ (%)	E_{ss} (Hz)	$\Delta\tau_{max}$ ($\%\tau_{nom}$)
"Motor Real"	**31**	**1118**	**0.49**	**1.44**	**0.00**	**359.0**
CASE D	36	1134	0.83	0.78	0.48	362.7
MD (Ours)	30	1133	0.69	0.52	0.45	355.4

TABLE VI: Performance metrics for the online inference on RDynamic-Speed1 benchmark for 1.5kW motor.

Table V shows the aggregated ML metrics on the dynamic and quasi-static real benchmarks obtained by DiagBiRNN with MD and in Case D. It confirms that DiagBiRNN-MD performs very well compared to Case D owing to its denoising capability. Table VI shows electrical engineering metrics for RDynamic-Speed1 benchmark. It should be pointed out that, for consistency, electrical engineering metrics for Case D are computed on smooth reconstruction of Case D predictions. Moreover, "Motor Real" stands for "non-noisy (reconstructed)", which has been obtained manually by doing an a posteriori approximation of the non-noisy trajectory. It can be seen that electrical engineering metrics of both Case D and DiagBiRNN-MD predictions are very close to real data metrics, DiagBiRNN-MD being even a little closer.

Figure 9 shows results from MD and Case D on this benchmark. Sub-figures (a) and (c) show speed-torque estimation from Case D and MD-DiagBiRNN, while sub-figures (b) and (d) show speed and torque prediction errors with respect to non-noisy (reconstructed) real speed and torque. Unlike DiagBiRNN-MD predictions, both speed and torque predictions for Case D are corrupted with a level of noise comparable to the noisy (measured) real data. This is confirmed by sub-figures (b) and (d), which show that case D predictions are pretty good in average but highly varying, while DiagBiRNN-MD predictions are not only clean but also very close to the non-noisy (reconstructed) real signals. Figure 10 shows results for RDynamic-Speed2 benchmark which contains multiple speed inversions in both directions. It can be observed that the MD-DiagBiRNN performs significantly better than case D predictions on such a challenging benchmark.

Method	$t_{95\%}$ (ms)	$D\%$ (%)	E_{ss} ($\%\tau_{nom}$)	SD (Hz)
"Motor Real"	**370**	**12.8**	**-0.4**	**3.41**
CASE D	373	13.2	-0.2	3.56
MD (Ours)	372	12.9	-0.3	3.45

TABLE VII: Performance metrics for the predictions done on real data RDynamic-Torque benchmark

Figure 11 shows results obtained on RDynamic-Torque benchmark. Case D predictions have significant amount of noise, whereas DiagBiRNN-MD prediction error is close to 0. In Table VII, reporting electrical engineering metrics for RDynamic-Torque benchmark, it can be seen

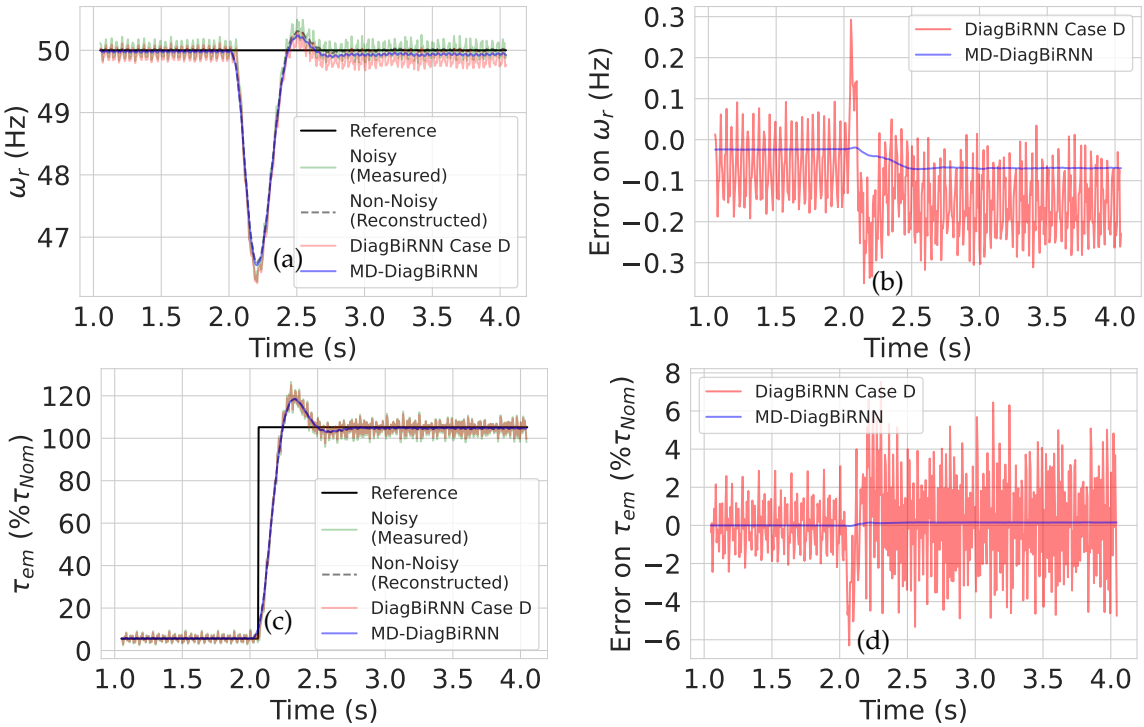


Fig. 11: Speed-torque estimator (DiagBiRNN) performance on RDynamic-Torque real data benchmark, with transfer learning (Case D) or with our proposed meta-denoiser (MD)

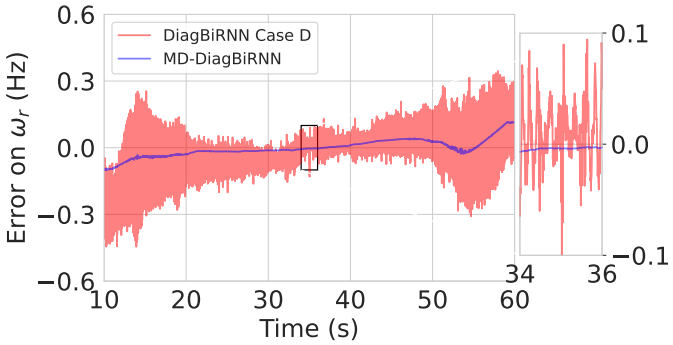


Fig. 12: Results on RQuasi-Static benchmark.

that, as for the previous benchmark, electrical engineering metrics of both DiagBiRNN-MD and smoothed Case D predictions are very close to real data metrics.

Figure 12 displays results on RQuasi-Static benchmark for Case D and DiagBiRNN-MD. As for the dynamic benchmarks, both methods perform well in average but with some noticeable noise for Case D prediction. The difference between prediction with DiagBiRNN-MD and non-noisy (reconstructed) real speed is never greater than 0.1Hz, which is very good.

C. Discussion

All the results obtained on the simulated and real benchmarks show that MD-DiagBiRNN outperforms other methods in estimating non-noisy speed and torque from noisy currents and voltages. Results on the four real data benchmarks show that both Case D and DiagBiRNN-MD provide significantly better speed and torque prediction. Case D predicts noisy speed and torque due to the fact that it is fine-tuned on train set of real data. DiagBiRNN-MD predicts speed and torque very close to the ideal (reconstructed) signals owing to the good denoising performance of the proposed MD technique. MD outperforms all other methods since it is not biased to static parts in the input noisy signals. Other methods perform relatively poorly in dynamic parts. This justifies the need for MD to be trained to perform segmentation and denoising simultaneously.

VI. Conclusions

We have developed a data-driven approach for estimating speed and torque from measured currents and voltages of an induction motor. This method allows us to make the bridge between data simulated from a physical model and real-world ones. We showed however that standard techniques for learning the underlying dynamical model are prone to errors in the presence of measurement noise. To overcome such shortcomings, we proposed a novel meta-denoiser (MD) method that removes noise from currents and voltages before feeding them to our speed-torque estimator (DiagBiRNN). We showed that the proposed approach performs very well on real data benchmarks. This neural estimator paves the way for fault detection algorithms for induction motors and their applications, by comparing speed and torque predictions to measured data. An interesting future work would be to explore the proposed method to estimate speed and torque using a motor soft starter, for which access to a precise motor model is more difficult than for a VSD.

REFERENCES

[1] S. J. Campbell, *Solid-State AC Motor Controls*, 1987.

[2] B. Weilharter, O. Biro, H. Lang, G. Ofner, and S. Rainer, "Validation of a comprehensive analytic noise computation method for induction machines," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 5, pp. 2248–2257, 2012.

[3] A. K. Jebai, P. Combes, F. Malrait, P. Martin, and P. Rouchon, "Energy-based modeling of electric motors," in *53rd IEEE Conference on Decision and Control*, pp. 6009–6016, Dec. 2014.

[4] P. J. Nicklasson, R. Ortega, G. Espinosa-Perez, and C. G. J. Jacobi, "Passivity-based control of a class of Blondel-Park transformable electric machines," *IEEE Transactions on Automatic Control*, vol. 42, no. 5, pp. 629–647, May. 1997.

[5] R. Marino, S. Peresada, and P. Tomei, "Global adaptive output feedback control of induction motors with uncertain rotor resistance," *IEEE Transactions on Automatic Control*, vol. 44, no. 5, pp. 967–983, May. 1999.

[6] M. Habibullah and D. D.-C. Lu, "A speed-sensorless FS-PTC of induction motors using extended Kalman filters," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 11, pp. 6765–6778, 2015.

[7] E. Zerdali and M. Barut, "The comparisons of optimized extended Kalman filters for speed-sensorless control of induction motors," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 6, pp. 4340–4351, 2017.

[8] A. Giantomassi, F. Ferracuti, S. Iarlori, G. Ippoliti, and S. Longhi, "Electric motor fault detection and diagnosis by kernel density estimation and Kullback-Leibler divergence based on stator current measurements," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 3, pp. 1770–1780, 2015.

[9] M. Yang, N. Chai, Z. Liu, B. Ren, and D. Xu, "Motor speed signature analysis for local bearing fault detection with noise cancellation based on improved drive algorithm," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 5, pp. 4172–4182, 2020.

[10] S. M. Gadoue, D. Giaouris, and J. W. Finch, "Sensorless control of induction motor drives at very low and zero speeds using neural network flux observers," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 8, pp. 3029–3039, 2009.

[11] T. Boukra, A. Lebaroud, and G. Clerc, "Statistical and neural-network approaches for the classification of induction machine faults using the ambiguity plane representation," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 9, pp. 4034–4042, 2013.

[12] S. Verma, N. Henwood, M. Castella, F. Malrait, and J.-C. Pesquet, "Modeling electrical motor dynamics using encoder-decoder with recurrent skip connection," in *AAAI conference on artificial intelligence*, pp. 1–8, 2020.

[13] S. Verma, N. Henwood, M. Castella, A. K. Jebai, and J.-C. Pesquet, "Neural networks based speed-torque estimators for induction motors and performance metrics," in *IECON 2020 - 46th Annual Conference of the IEEE Industrial Electronics Society*, pp. 495–500, 2020.

[14] Y. Liu, "Noise reduction by vector median filtering," *GEOPHYSICS*, vol. 78, no. 3, pp. V79–V87, 2013.

[15] B. Anderson and J. Moore, *Optimal Filtering*. Prentice-Hall, 1979.

[16] Ã. P. Dautov and M. S. Özerdem, "Wavelet transform and signal denoising using wavelet method," in *2018 26th Signal Processing and Communications Applications Conference*, pp. 1–4, 2018.

[17] K. Naveed, B. Shaukat, and N. U. Rehman, "Dual tree complex wavelet transform-based signal denoising method exploiting neighbourhood dependencies and goodness-of-fit test," *Royal Society Open Science*, vol. 5, no. 9, 2018.

[18] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: Nonlinear Phenomena*, vol. 60, no. 1, pp. 259–268, 1992.

[19] I. Selesnick, "Total variation denoising via the Moreau envelope," *IEEE Signal Processing Letters*, vol. 24, no. 2, pp. 216–220, 2017.

[20] H. Du and Y. Liu, "Minmax-concave total variation denoising," *Signal, Image and Video Processing*, vol. 12, 09 2018.

[21] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, pp. 3371–3408, 2010.

[22] N. Henwood, J. Malaizé, and L. Praly, "PMSM identification for automotive applications: Cancellation of position sensor errors," in *IECON 2011 - 37th Annual Conference of the IEEE Industrial Electronics Society*, pp. 687–692, 2011.

[23] K. R. Muske and T. F. Edgar, *Nonlinear State Estimation*, pp. 311–370. Prentice-Hall, 1997.

[24] H. A. Chipman, E. D. Kolaczyk, and R. E. McCulloch, "Adaptive Bayesian wavelet shrinkage," *Journal of the American Statistical Association*, vol. 92, no. 440, pp. 1413–1421, 1997.

[25] E. Zerdali, "Adaptive extended kalman filter for speed-sensorless control of induction motors," *IEEE Transactions on Energy Conversion*, vol. 34, no. 2, pp. 789–800, 2019.

[26] K. P. Murphy, *Probabilistic Machine Learning: An introduction*. MIT Press, 2021. [Online]. Available: probml.ai

[27] S. Verma and J.-C. Pesquet, "Sparsifying networks via subdifferential inclusion," in *Proceedings of the 38th International Conference on Machine Learning*, pp. 10 542–10 552, 2021.

Sagar Verma received the B.Tech degree from NIT Raipur, India, in 2014 and the M.Tech from IIIT Delhi, India, in 2017. He was Research Engineer at IBM Research Lab India from 2017 to 2018. He is currently pursuing his Ph.D. at CentraleSupélec, Université Paris-Saclay with Schneider Toshiba Inverter Europe, France.

Nicolas Henwood received his engineering degree from Supélec, Gif-sur-Yvette, France, in 2010 and his Ph.D. in Mathematics and Control from Mines ParisTech, Paris, France, in 2014. From 2010 to 2014, he worked jointly between IFP Energies Nouvelles and the Centre Automatique et Systèmes from Mines ParisTech. Since 2014, he works as a Motor Control engineer at the Techno Center Control Systems of Schneider Electric, France.

Marc Castella (Member, IEEE) received the Agrégation degree in the field of applied physics in 2000. Then, he received the M.Sc. degree in electrical engineering, from the École Normale Supérieure de Cachan, France. Finally, he received the Ph.D. degree from the Université de Marne-la-Vallée, France, in 2004. Since then, he has been Assistant Professor in Evry, France, at Télécom SudParis.

Al Kassem Jebai received the Diploma of engineer from Mines ParisTech, Paris, France, in 2009, and the Ph.D. degree in automatica and applied mathematics from Mines ParisTech, in 2013. In 2014, he joined the Motor Control Team, Schneider Toshiba Inverter Europe. Since 2019, he has been the Head of Technology Center Control Systems for Drives activity of Schneider Toshiba Inverter Europe.

Jean-Christophe Pesquet (Fellow, IEEE 2012) received the engineering degree from Supélec, Gif-sur-Yvette, France, in 1987, the Ph.D. and HDR degrees from Université Paris-Sud in 1990 and 1999, respectively. He is currently a Distinguished Professor at CentraleSupélec, Université Paris-Saclay, and the Director of the Center for Visual Computing and OPIS Inria group. He has also been a Senior Member of the Institut Universitaire de France since 2016.