Neural Networks based Speed-Torque Estimators for Induction Motors and Performance Metrics

S. Verma[1,2] N. Henwood[2] M. Castella[3] A. K. Jebai[2] J.C. Pesquet[1]

[1]Université Paris-Saclay, CentraleSupélec, Inria, Centre de Vision Numérique

[2]Schneider Toshiba Inverter Europe

[3]SAMOVAR, Télécom SudParis, Institut Polytechnique de Paris

46th Annual Conference of the IEEE Industrial Electronics Society

Table of Contents

Table of Contents

Introduction

Evaluating neural networks (NNs) used in induction motor use case using performance metrics.

- Physics for dynamics modeling.
- ML methods for control and fault detection.
- Large industrial sensor datasets.
- Performance metrics to evaluate NN.

Problem Statement

Estimating speed and torque from current and voltage of an induction motor using NNs.

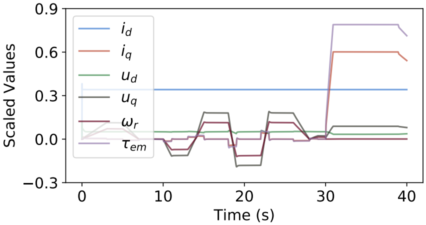


Figure: First 40 seconds of a simulated electrical motor operation.

Table of Contents

Physical Modeling

Model based controller requires modeling dynamics.

- Controller design with known parameters [*Nicklasson et al., 1997*].
- Controller design with uncertain parameters [*Marino et al., 1999*].
- State-space model of induction motor [*Jadot et al., 2009*].
- Analytical mechanics and energy consumption [*Jebai et al., 2014*].

Neural Network based Modeling

Control and fault detection using NNs.

- NN fault classifiers [*Silva et al., 2013*].
- Currents and flux linkages coupling using RBF [*Ortombina et al., 2018*].
- Encoder-decoder learning input-output relationship [*Verma et al., 2020*].

Table of Contents

Reference Trajectory Generator

- Generate reference speed and torque load
- Real world operating scenarios
- Conditional Hidden Markov model
- Paramtereized operating conditions

Reference Trajectory Generator

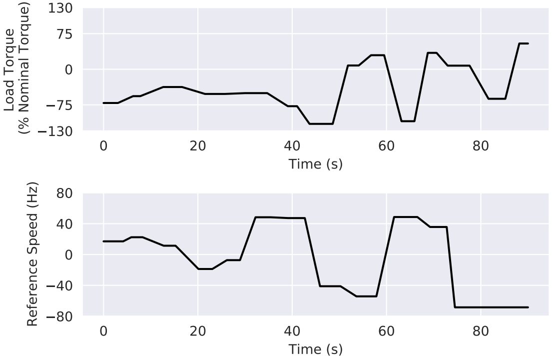


Figure: Generated reference speed and load torque trajectories.

Nonlinear State-Space Motor Model

- Generate simulation data.
- Fifth-order nonlinear state space model [*Jadot et al., 2009*]
- Widely used in industrial paradigm.
- Explained in **Physical Modeling: A**.

Standard Neural Networks

Three standard networks from [*Verma et al., 2020*]

- Four layer Fully Connected Network (**FCN**)
- Two layer Long-Short Term Memory Network (**LSTM**)
- Four layer Convolutional Neural Network (**CNN**)

Four layer Fully Connected Network (FCN)

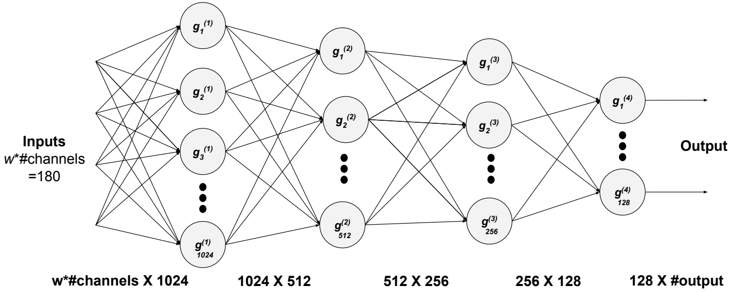


Figure: Fully Connected Network.

Two layer Long-Short Term Memory Network (LSTM)

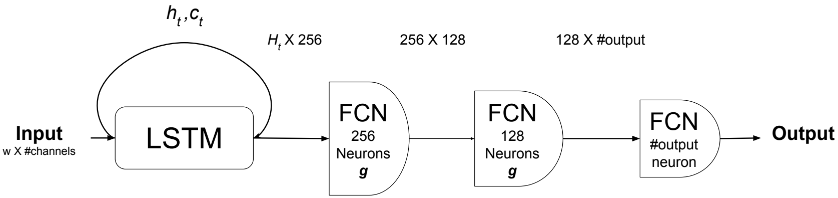


Figure: Long-Short Term Memory Network.

Four layer Convolutional Neural Network (CNN)

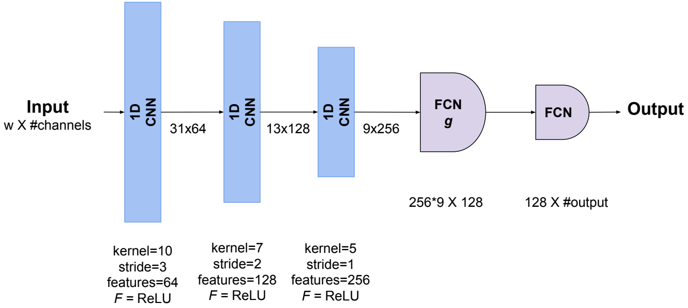


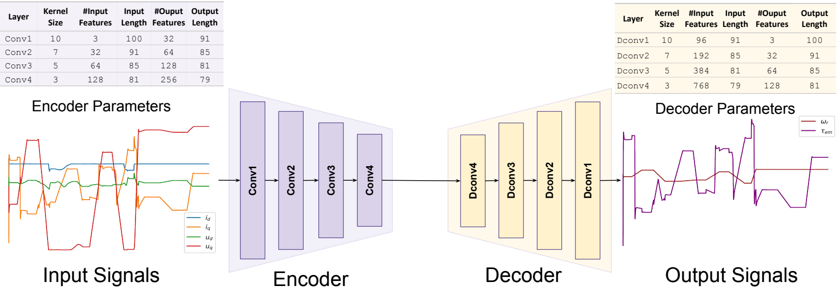
Figure: Convolutional Neural Network.

Encoder-Decoder Networks

Proven performance gain in temporal signal modeling.

- Vanilla Encoder-Decoder (**Vanilla**)
- Encoder-Decoder with Skip Connections (**Skip**)
- Encoder-Decoder with Recurrent Skip Connections (**RNN**)
- Encoder-Decoder with Bidirectional Recurrent Skip Connections (**BiRNN**)
- Encoder-Decoder with Bidirectional Diaganolized Recurrent Skip Connections (**DiagBiRNN**)

Vanilla Encoder-Decoder (Vanilla)

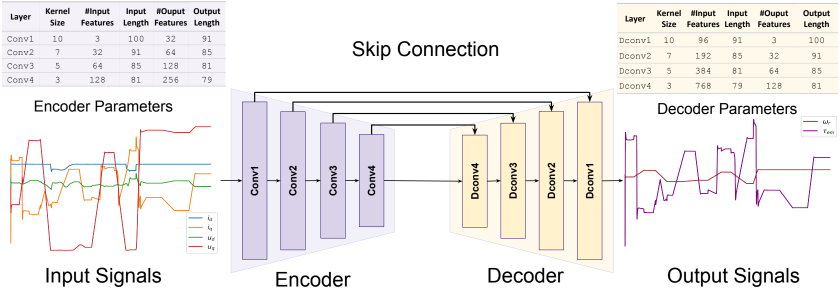


Layer	Kernel Size	#Input Features	Input Length	#Ouput Features	Output Length
Conv1	10	3	100	32	91
Conv2	7	32	91	64	85
Conv3	5	64	85	128	81
Conv4	3	128	81	256	79

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

Figure: Vanilla

Encoder-Decoder with Skip Connections (Skip)



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

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

Figure: Skip Connections

Encoder-Decoder with Recurrent Skip Connections (RNN)

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

Encoder Parameters

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

Decoder Parameters

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

Recurrent Skip Connection Parameters

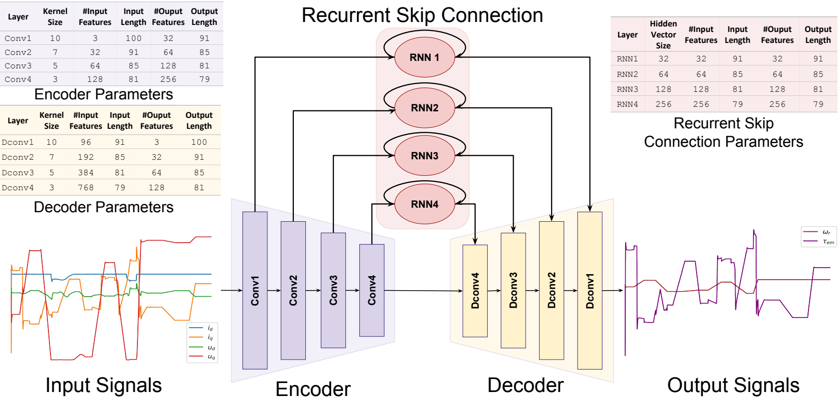


Figure: Recurrent Skip Connections

Encoder-Decoder with Bidirectional Recurrent Skip Connections (BiRNN)

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

Encoder Parameters

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

Decoder Parameters

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

Recurrent Skip Connection Parameters
All RNNs are bidirectional

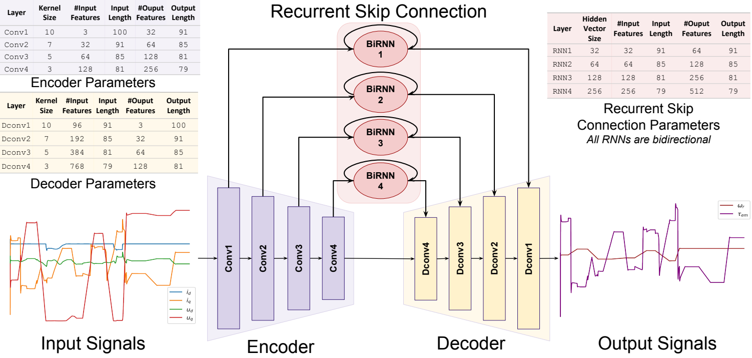
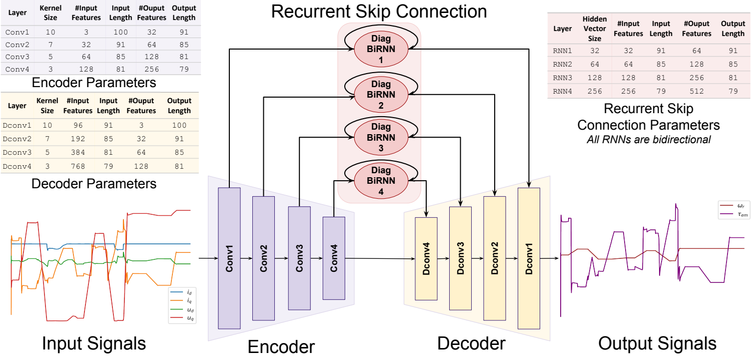


Figure: Bidirectional Recurrent Skip Connections

Encoder-Decoder with Bidirectional Diaganolized Recurrent Skip Connections (DiagBiRNN)



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

Encoder Parameters

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

Decoder Parameters

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

Recurrent Skip Connection Parameters
All RNNs are bidirectional

Figure: Bidirectional Diaganolized Recurrent Skip Connections

Analyze neural network performance using ML metrics.

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

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

$$R^2(y,\hat{y}) = 1 - \frac{\sum_{t=1}^{T}(\hat{y}_t - \bar{y})^2}{\sum_{t=1}^{T}(y_t - \bar{y})^2}$$

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

Performance Metrics

Electrical engineering (EE) performance metrics widely used in industrial settings.

- **2% response time** ($t_{2\%}$)
- **95% response time** ($t_{95\%}$)
- **Overshoot** ($D\%$)
- **Steady-state error** (E_{ss})
- **Following error** (E_{fol})
- **Maximum acceleration torque (for speed ramp only)** ($\Delta\tau_{max}$)
- **Speed drop (for torque ramp only)** (SD)

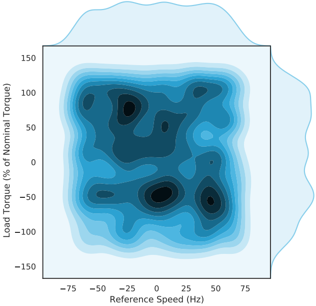
CentraleSupélec Schneider Electric TELECOM SudParis IP PARIS

Table of Contents

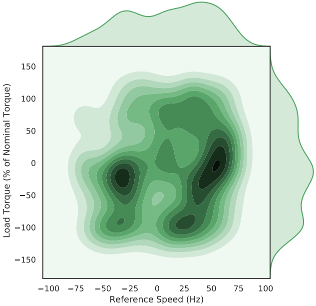
Training and Validation Dataset Generation

- **100** simulated speed and torque trajectories.
- State-space model to simulate.
- Different speed and torque ramps from exponential distribution.
- **4kW** induction motor at **4kHz**.
- **150 minutes** of training and **30 minutes** of validation.

Training and Validation Dataset Generation



(a) Training zone



(b) Validation zone

Figure: Density plots of torque vs speed plans showing training and validation separation.

Ramps Distribution Matters

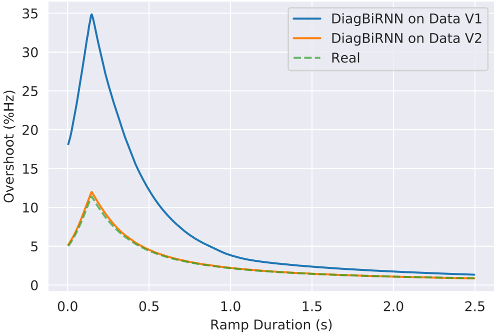


Figure: Overshoot vs. ramp. Bias in **Data V1** due to normal distribution. **Data V2** from exponential distribution.

Test Dataset Generation

Bench-marking NN methods using EE performance metrics.

- **Dynamic-Speed1**: 0 to 50Hz in 1s at no load.
- **Dynamic-Speed2**: 50 to -50Hz in 1s at 50% of nominal load.
- **Dynamic-Torque**: Load from 0 to 100% of nominal in 4ms at 25Hz.
- **Quasi-Static1**: No load, 70 to -70Hz in 50s.
- **Quasi-Static2**: 50% nominal load, 70 to -70Hz in 50s.

Experimental Setup

- Ubuntu 18.04 OS, V100 GPU, and PyTorch.
- **Simulink** for state-space model.
- **100** time steps is the input size for all networks.
- Learning rate is **0.1** for standard and **0.001** for encoder-decoder.
- Total variation weighted mean square loss $\mathscr{L}_{\text{TV-MSE}}$.

CentraleSupélec Schneider Electric TELECOM SudParis IP PARIS

Table of Contents

Machine Learning Metrics Aggregated

Model	Speed (ω_r)		Torque (τ_{em})	
	MAE	**SMAPE**	**MAE**	**SMAPE**
FCN	0.79	21.77%	0.57	48.66%
LSTM	0.11	18.76%	0.21	43.01%
CNN	0.06	19.14%	0.09	38.91%
R^2 is 0.99 for all the networks for both quantities.				

Table: ML metrics for the predictions done on benchmark set using standard models.

Machine Learning Metrics Aggregated

Model	Speed (ω_r)		Torque (τ_{em})	
	MAE	**SMAPE**	**MAE**	**SMAPE**
Vanilla	0.05	18.94%	0.10	39.91%
Skip	0.08	19.08%	0.12	43.23%
RNN	0.06	19.31%	0.08	41.81%
BiRNN	0.05	**18.67%**	0.09	42.82%
DiagBiRNN	**0.03**	18.76%	**0.04**	**38.46%**
R^2 is 0.99 for all the networks for both quantities.				

Table: ML metrics for the predictions done on benchmark set using the encoder-decoder variants.

CentraleSupélec Schneider Electric TELECOM SudParis IP PARIS

Performance Metrics on Dynamic Benchmarks

Model	$t_{2\%}$ (ms)	$t_{95\%}$ (ms)	E_{fol} (Hz)	$D\%$ (%)	E_{ss} (Hz)	$\Delta\tau_{max}$ ($\%\tau_{nom}$)
Real	**48**	**960**	**-0.02**	**2.16**	**0.00**	**32.69**
Vanilla	44	968	-0.08	2.62	0.02	32.37
Skip	**48**	952	0.12	3.04	**0.01**	32.46
RNN	**48**	952	-0.04	2.28	0.02	32.82
BiRNN	44	944	-0.11	2.29	**0.01**	**32.67**
DiagBiRNN	44	952	**-0.01**	**2.21**	0.03	**32.67**

Table: EE performance metrics obtained by encoder-decoder networks on Dynamic-Speed1 benchmark.

Performance Metrics on Dynamic Benchmarks

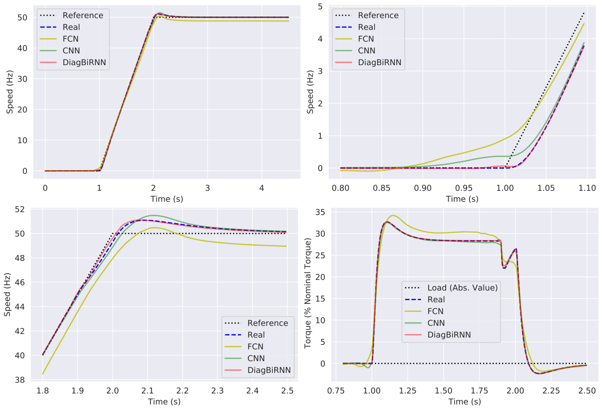


Figure: Results on Dynamic-Speed1 benchmark.

Performance Metrics on Dynamic Benchmarks

Model	$t_{95\%}$ (ms)	$D\%$ (%)	E_{ss} ($\%\tau_{nom}$)	SD (Hz)
Real	**244**	**15.96**	**0.00**	**4.39**
Vanilla	**244**	15.46	0.04	3.88
Skip	**244**	15.90	-0.02	**4.43**
RNN	**244**	15.87	**0.00**	4.03
BiRNN	**244**	16.01	0.01	4.23
DiagBiRNN	**244**	15.91	-0.02	4.31

Table: EE performance metrics obtained by different models on Dynamic-Torque benchmark.

Performance Metrics on Dynamic Benchmarks

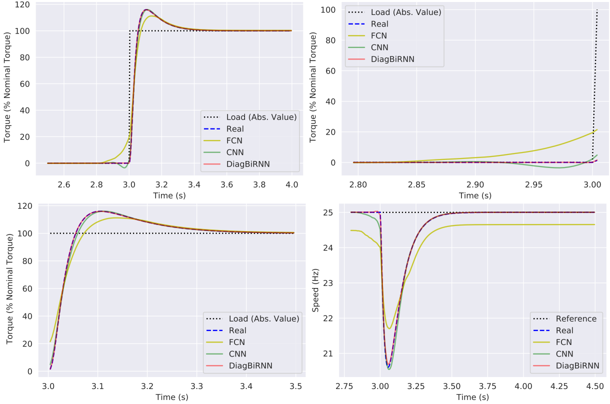


Figure: Results on Dynamic-Torque benchmark.

Performance Metrics on Static Benchmarks

FCN	LSTM	CNN	Vanilla	Skip	RNN	BiRNN	DiagBiRNN
3.66	0.992	0.261	0.178	0.549	0.341	0.236	**0.198**

Table: Max absolute error (Hz) for Quasi-Static1 benchmark.

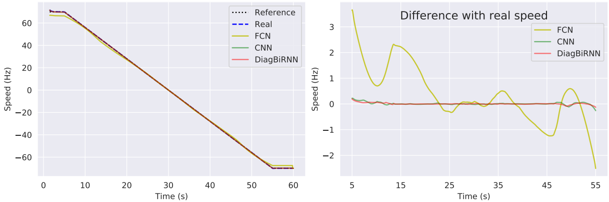


Figure: Results on Quasi-Static1 benchmark.

CentraleSupélec

Schneider Electric

TELECOM SudParis

IP PARIS

Conclusion

- NNs for speed-torque estimation are not trivial.
- Special care in dataset generation and training procedure.
- ML metrics gives false sense performance gain.
- EE performance metrics are best suited for this kind of problems.

Thanks!

Contact: sagar.verma@se.com

Figure: Project page, code, and dataset.