Surrogate Modelling of Dynamic Phasor Simulations of Electrical Drives

Nasrulloh Ratu Bagus Satrio Loka
INTEC
Ghent University - IMEC
Ghent, Belgium
nasruloh.loka@ugent.be

Sriram Karthik Gurumurthy
ACS, EONERC
RWTH Aachen University
Aachen, Germany
sgurumurthy@eonerc.rwth-aachen.de

Bernard Amevor
ACS, EONERC
RWTH Aachen University
Aachen, Germany
bernard.amevor@rwth-aachen.de

Antonello Monti
ACS, EONERC
RWTH Aachen University
Aachen, Germany
amonti@eonerc.rwth-aachen.de

Tom Dhaene
INTEC
Ghent University - IMEC
Ghent, Belgium
tom.dhaene@ugent.be

Ivo Couckuyt
INTEC
Ghent University - IMEC
Ghent, Belgium
ivo.couckuyt@ugent.be

Abstract—**This work develops and benchmarks surrogate models for Dynamic Phasor (DP) simulation of electrical drives. DP simulations of complex systems may be time-consuming due to the increased number of equations. Thus, it is desirable to have a data-driven approach to compute the critical state/control variables and power losses. The surrogate models are intended to be used as a steady-state equivalent of the DP simulation model. We consider the Gaussian Process (GP), Multi Layer Perceptron, and Random Forest as surrogate models. Among other techniques, GPs are found to have good accuracy. Moreover, GPs are data-efficient and have desirable properties, such as built-in uncertainty quantification. The study shows that the GP performs better compared to other techniques in terms of the Mean Squared Error of the prediction, while still being very fast to evaluate. We illustrate the potential of these surrogate models to also predict transient behavior.**

Index Terms—**Dynamic phasor, Induction machine, Permanent magnet synchronous motor, Surrogate modelling, Gaussian Process**

I. INTRODUCTION

Conventionally, Electromagnetic-Transient (EMT) simulations and steady-state phasor simulations are widely used for simulating power grids. EMT simulations are suitable for switched power converters at the cost of increased computation and simulation time and steady-state phasor simulations are extremely fast but they are not suitable for transient studies. Dynamic Phasor (DP) combines the advantages of both EMT and steady-state phasor methods by performing frequency-shift operation on the original time-domain signal. This results in a complex signal that captures the envelope of the original time-domain signal. The DP method allows a larger time step and allows selective modelling of dominant frequencies. However,

for complex systems, due to the increased number of equations, the simulation time may increase. In this study, we investigate the performance of several data-driven surrogate models which are built upon the data set recorded from the DP simulation.

Recently, surrogate models have been applied to power electronic converters focusing on the thermal modelling of power electronic devices [1], [2]. The input power and simulation time are considered as input parameters and they are mapped to the junction temperature [1]. In [2], an algorithm is developed to extract parameters of thermal models. Furthermore, in [3], surrogate models were developed for reliability and lifetime analysis of power electronics.

In a previous study [1], a neural network (NN) is used, however, NNs need a lot of data to perform well. Secondly, NN hyperparameter selection is hard. Finding the right architecture per problem is a time-consuming process. Considering those drawbacks, we benchmarks several surrogate modelling techniques namely: the Multi Layer Perceptron (MLP) [4], Gaussian Process (GP) [5], [6], [7], and Random Forest (RF) [8]. MLP is one of the standard variants of NNs for a regression problem. GPs could be a suitable alternative to NNs, as GPs are data-efficient, able to use prior knowledge via the choice of the kernel function, and also it can capture the model uncertainty when seeing new data. The latter could be useful for decision making, active learning, or anomaly detection. Furthermore, unlike NNs; GPs do not need time-consuming hyperparameter tuning.

We consider two motor types for the surrogate models: Induction Motor (IM), and Permanent Magnet Synchronous Motor (PMSM) drives based on DP simulations. The surrogate models are built to map the input variables such as DC link voltage, load torque, and speed reference to the output variables of the drive including stator currents, modulation ratio, and modulation angle. Furthermore, the input variables are also mapped to the losses in the drive. We consider the mechanical and electrical losses for the motor. For the three-phase inverter,

we consider the conduction and switching losses. The surrogate model can be used as the steady-state equivalent of the dynamic simulation model. Further, we also demonstrate the potential usage of the surrogate model to predict the transient behaviour.

II. DP MODELLING OF ELECTRICAL COMPONENTS
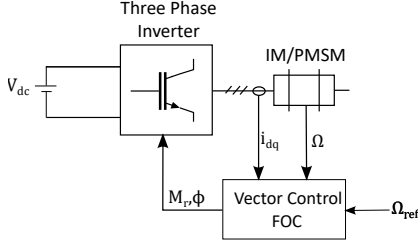
A. Electrical System Description

Fig. 1: Electrical system layout

The model of the electrical system considered is shown in Figure 1. The system consists of a three-phase 2-level converter that interfaces a DC power supply and an AC electrical motor. In this paper, we have considered two motor types: a three-phase squirrel cage Induction Motor (IM) and a Permanent Magnet Synchronous Motor (PMSM). A suitable Field Oriented Control (FOC) strategy is used for both types of motor to track the desired mechanical reference speed of the motor by actively controlling the output voltage vectors of the three-phase inverter.

B. Dynamic Phasor (DP) Modelling Approach

Dynamic phasors are considered for modelling the electrical components. The dynamic phasors are the time-dependent Fourier coefficient of a periodic signal with period T and they are defined as follows:

$$\langle v \rangle_0(t) = V_{dc}(t) \tag{1}$$

$$\langle v \rangle_k(t) = \frac{1}{T} \int_{\tau-T}^{\tau} v(t) e^{-jk\omega_o t} d\tau \tag{2}$$

$$\frac{d}{dt} \langle v \rangle_k(t) = \left\langle \frac{d}{dt} v \right\rangle_k (t) - j\omega_o k \langle \psi \rangle_k(t) \tag{3}$$

The product of two dynamic phasors, v and i is calculated through the discrete convolution principle given by (4).

$$\langle vi \rangle_k = \sum_{l=-\infty}^{\infty} \langle v \rangle_{k-l} \langle i \rangle_l \tag{4}$$

Considering that the electrical power flowing in the system is primary caused by the DC component of voltage/current and the fundamental 50 Hz component of voltage/current, the power definition as per (4) can be reduced to the following:

$$\langle p \rangle_0 = \langle v \rangle_0 \langle i \rangle_0 + 2\Re\left\{ \langle v \rangle_1 \langle i \rangle_{-1} \right\}$$

$$\langle q \rangle_0 = 2\Im\left\{ \langle v \rangle_1 \langle i \rangle_{-1} \right\}$$

AC electrical machines are typically modelled in the Direct−Quadrature (DQ) domain [9]. The stationary domain phasors v_{abc} can be transformed into positive sequence DQ domain phasors using the transformation in (5), where $\alpha = e^{j2\frac{\pi}{3}}$ [9].

$$\langle V_p \rangle_k = \frac{2}{3} \begin{bmatrix} 1 & \alpha & \alpha^2 \end{bmatrix} \begin{bmatrix} \langle v_a \rangle_k \\ \langle v_b \rangle_k \\ \langle v_c \rangle_k \end{bmatrix} \tag{5}$$

C. Induction Machine (IM) Model

The stator flux λ_{ps} and rotor flux λ_{pr} depend on the stator currents i_{ps} and rotor currents i_{pr} as shown below:

$$\begin{bmatrix} \lambda_{ps} \\ \lambda_{pr} \end{bmatrix} = \begin{bmatrix} L_s & L_m \\ L_m & L_r \end{bmatrix} \begin{bmatrix} i_{ps} \\ i_{pr} \end{bmatrix} \tag{6}$$

Here, L_s, L_r and L_m represent the stator self-inductance, rotor self-inductance and mutual inductance between stator and rotor respectively.

1) Stator Equations: The stator currents are driven by the applied stator voltage v_{ps} and electrical supply angular frequency ω_s.

$$\frac{d\lambda_{ps}}{dt} = V_{ps} - r_s i_{ps} - j\omega_s \lambda_{ps} \tag{7}$$

In the above equations, r_s represents the equivalent stator resistance.

2) Rotor Equations: Due to the squirrel cage rotor assumption, there are no externally applied voltages. The rotor currents are driven by the flux from the stator. r_r represents the equivalent rotor resistance and ω_{slip} represents the slip frequency.

$$\frac{d\lambda_{pr}}{dt} = -r_r i_{pr} - j\omega_{slip} \lambda_{pr,im} \tag{8}$$

3) Torque and Angular Velocity Equations: The electrical torque T_e developed by the IM and the mechanical angular speed Ω_m equation obtained by torque balance is given below:

$$T_e = \frac{3}{2} p (\lambda_{ps,re} i_{ps,im} - \lambda_{ps,im} i_{ps,re}) \tag{9}$$

$$J \frac{d\Omega_m}{dt} = T_e - B\Omega_m - T_L \tag{10}$$

In the above equations, p represents the number of pole pairs in the stator, B is the coefficient of friction and T_L is the load torque. The slip frequency in IM is defined as follows:

$$\Omega_{el} = p\Omega_m \tag{11}$$

$$\omega_{slip} = \omega_s - \Omega_{el} \tag{12}$$

4) Field Oriented Control: In FOC, the rotor flux is held constant in a particular orientation $\lambda_{pr,re} = \Lambda_o$ and $\lambda_{pr,im} = 0$ which allows a linear relationship between the required electrical torque and current. An external speed controller generates the torque command as shown in Figure 2. The imaginary axis current contributes to the torque developed, thus depending on the torque, the current reference is calculated according to (13).

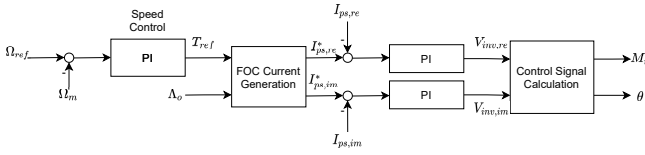$$i_{ps,im}^* = \frac{T_e^*}{1.5p\frac{L_m}{L_r}\Lambda_o} \tag{13}$$

Fig. 2: FOC in IM and PMSM

The stator real axis current reference is calculated depending on the required rotor flux as shown in (14).

$$i_{ps,re}^* = \frac{\Lambda_o}{L_m} \tag{14}$$

The required stator frequency is calculated using (15).

$$\omega_{s,est} = p\Omega_m + \underbrace{\frac{r_r L_m i_{pr,im}}{L_r \Lambda_o}}_{\omega_{slip,est}} \tag{15}$$

A Synchronous Reference Frame (SRF) current controller with decoupling is used to track the generated reference current. The output of the PI controller is the inverter control voltage which is transformed into the modulation ratio M_r and angle ϕ based on the available DC link voltage V_{dc} :

$$M_r = \frac{\sqrt{V_{inv,re}^2 + V_{inv,im}^2}}{\frac{V_{dc}}{2}}$$

$$\phi = \arctan\left(\frac{V_{inv,im}}{V_{inv,re}}\right)$$

D. Permanent Magnet Synchronous Machine (PMSM) Model

Unlike the IM, the stator flux λ_{ps} in PMSM depends only on the stator currents i_{ps} since there is no electrical circuit in the rotor [10]. The stator flux real part depends on the permanent magnet flux λ_f from the rotor.

$$\begin{bmatrix} \lambda_{ps,re} \\ \lambda_{ps,im} \end{bmatrix} = \begin{bmatrix} L_{sd} & 0 \\ 0 & L_{sq} \end{bmatrix} \begin{bmatrix} i_{ps,re} \\ i_{ps,im} \end{bmatrix} + \begin{bmatrix} \lambda_f \\ 0 \end{bmatrix} \tag{16}$$

L_{sd} and L_{sq} represent the direct and quadrature axis self-inductance.

1) Stator Equations: The stator currents are driven by the applied stator voltage v_{ps} and electrical supply angular frequency ω_s.

$$\frac{d\lambda_{ps}}{dt} = V_{ps} - r_s i_{ps} - j\omega_s \lambda_{ps} \tag{17}$$

2) Torque and Angular Velocity Equations: The electrical torque T_e developed by the PMSM is given below. The torque balance equation is identical to (10).

$$T_e = \frac{3}{2} p(\lambda_f i_{ps,im} + \Delta L . i_{ps,re} i_{ps,im}) \tag{18}$$

3) Field Oriented Control: In interior PMSM, $L_{sd} > L_{sq}$, hence we define $\Delta L = L_{sd} - L_{sq}$. By Maximum Torque per Ampere (MTPA) strategy [10], the reference currents are calculated. The peak stator current I_s^* is calculated using the reference torque.

$$I_s^* = \frac{T_e^*}{1.5 p \lambda_f} \tag{19}$$

By applying the MTPA condition, the real axis current reference is calculated according to (20). Due to quadratic nature, two solutions are obtained from which the negative solution has to be chosen for motors.

$$i_{ps,re}^* = \frac{1}{4\Delta L}\left(-\lambda_f \pm \sqrt{\lambda_f^2 + 8\Delta L I_s^2}\right) \tag{20}$$

The imaginary axis current is calculated as shown below.

$$i_{ps,im}^* = \sqrt{I_s^{*2} - i_{ps,re}^{*2}} . \operatorname{sign} I_s^* \tag{21}$$

The required stator frequency is calculated using the mechanical speed $\omega_{s,est} = p\Omega_m$

E. Three-Phase Inverter Model

We assume a B6C topology for the three-phase inverter and it is assumed that the gate signal is generated through a sine pulse width modulation (SPWM) strategy. The three-phase inverter is modelled as a sinusoidal voltage source in the DP domain by neglecting harmonics. The dynamic phasor of the phase to neutral voltage of phase A v_{an} can be calculated as shown in (22) [11].

$$\langle v_{an} \rangle_0 = \frac{M_r V_{dc}}{4} e^{j\phi} \tag{22}$$

The voltage on the DC link is V_{dc} and the control variables are Modulation ratio M_r and phase-shift angle φ. Power electronic switches such as IGBTs and MOSFETs are typically used in motor drives, IGBTs are preferred for medium voltage applications due to their inherent capabilities to withstand higher voltages. Most power electronic topologies require an anti-parallel diode for commutation of the line current during a switching event.

Consider the voltage drop across the IGBT and diode during conduction as $V_{ce,S}$ and $V_{ce,D}$, respectively. The conduction loss in a three-phase 2-level converter is given by (23).

$$P_{cond} = 6\frac{|\langle i \rangle_0|}{\pi}\left\{V_{ce,S}\left(1 + \frac{M_r \pi cos\varphi}{4}\right)\right.$$
$$\left. + V_{ce,D}\left(1 - \frac{M_r \pi cos\varphi}{4}\right)\right\} \tag{23}$$

$|\langle i \rangle_0|$ represents the peak value of current through the switch. The turn-off and turn-on energy can be calculated at a given reference DC link voltage V_{ref} and reference peak current I_{ref}. The turn-off and turn-on energy of the IGBT are $E_{off,S}$ and $E_{on,S}$, respectively and the turn-off energy of the diode is E_D. For B6C topology, the switching losses across the 6 IGBT/diode pair is given by (24).

$$P_{sw} = 6\frac{f_{sw}}{\pi}\left(E_{on,S} + E_{off,S} + E_D\right)\frac{V_{dc}}{V_{ref}}\frac{2|\langle i \rangle_0|}{I_{ref}} \tag{24}$$

III. SURROGATE MODELS

Compact surrogate models are fast-to-evaluate and can be run on a multitude of platforms. We train and benchmark various surrogate models based on the DP simulations.

A. Data generation

The goal of the data generation process is to obtain informative data sets which describe the system best. In other words, we want space-filling designs that spread out the points to encourage a variety of data. The surrogates model trained on this data set is smooth within the specified domain and has better interpolation and extrapolation performance than data sets obtained through random sampling [12]. For this purpose, a Halton random sequence [13] is used. Compared to random sampling, Halton random sequences produce more dispersed points, see Figure 3, leading to better interpolation and extrapolation performance of the surrogate models. The boundary of the domain is not treated differently.
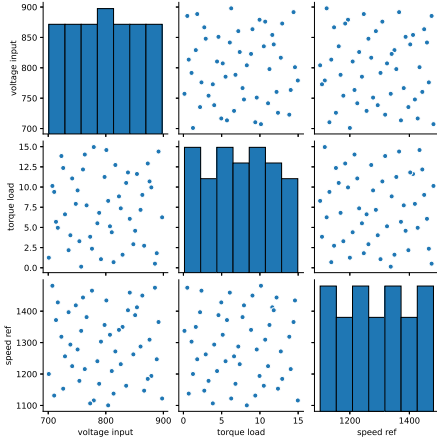

Fig. 3: Halton random sequence of 50 points for the IM case.

B. Gaussian Processes

A Gaussian Process (GP) can be viewed as a distribution over real-valued functions: $f(\mathbf{x}) \sim GP(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$. A GP is fully specified by its mean function $m(\mathbf{x})$ and covariance function $k(\mathbf{x}, \mathbf{x}')$ [5]. To train the GP model, we need to estimate the hyperparameters $\hat{\theta}$. In this case, $\hat{\theta}$ contain the parameters of $k(\mathbf{x}, \mathbf{x}')$. Maximum Likelihood Estimation (MLE) [14] is used to estimate the hyperparameters:

$$\hat{\theta} = \arg\max_{\theta} \log p(\mathbf{f} \mid X, \theta) \tag{25}$$

$$= \arg\max_{\theta} -\frac{1}{2}\left(\log|2\pi K_{\mathbf{xx}}| + \mathbf{f}^T K_{\mathbf{xx}}^{-1}\mathbf{f}\right) \tag{26}$$

where $X = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$ are the training data, $\mathbf{f}$ is a vector of function observations at X, and $K_{\mathbf{xx}} = k(\mathbf{x}_i, \mathbf{x}_j)$.

For predicting new points $X_\star = [\mathbf{x}_{\star 1}, \ldots, \mathbf{x}_{\star N_\star}]$, the Gaussian predictive distribution is given as:

$$\mu(X_\star) = K_{\star \mathbf{x}} K_{\mathbf{xx}}^{-1} \mathbf{y} \tag{27}$$

$$\sigma^2(X_\star) = K_{\star\star} - K_{\star \mathbf{x}} K_{\mathbf{xx}}^{-1} K_{\star \mathbf{x}}^T \tag{28}$$

where $\mu(X_\star)$ is the predictive mean, $\sigma^2(X_\star)$ is the predictive variance, $K_{\star \mathbf{x}} = k(\mathbf{x}_{\star i}, \mathbf{x}_j)$, and $K_{\star\star} = k(\mathbf{x}_{\star i}, \mathbf{x}_{\star j})$. For the covariance function, the Matérn $5/2$ kernel [15] is used:

$$k(\mathbf{x}, \mathbf{x}') = \alpha\left(1 + \sqrt{5}r + \frac{5}{3}r^2\right)\exp(-\sqrt{5}r), \tag{29}$$

$$r = \sqrt{\sum_{m=1}^{d} \frac{(x_m - x'_m)^2}{l_m^2}} \tag{30}$$

where α is the kernel variance, and l_m is the lengthscale for dimension m. This kernel is adopted since we assume that the physical model does not have overly-smooth response surfaces.

IV. SURROGATE MODEL SETTINGS

We consider four use cases to build the surrogate models, Induction Motor Simulation (IM-M), Permanent Magnet Synchronous Motor Simulation (PMSM-M), Three-Phase Inverter with IM Motor (IM-TPI), and Three-Phase Inverter with PMSM Motor (PMSM-TPI).

A. Data Sets

In this study, 250 points of training data are generated using the Halton random sequence[1]. The domain specification for the training data generation is given in the Table I.

TABLE I: Input domain specification

Parameter	Lower bound	Upper bound
DC Link Voltage	700 Volts	900 Volts
Load Torque	0 Nm	15 Nm
Desired Speed	1100 rpm	1490 rpm

The motor surrogate uses all of the three inputs described in Table I, while the three-phase inverter surrogate does not use the load torque variable and replaces it with the electrical input power instead. Note that the domain for the electrical input power variable is not fixed. Thus, its value will depend on the other variables. After generating the inputs points, we run the simulation[2] to obtain the output variables. We set the simulation time to 100 seconds with a 0.1 second time step to reach a steady-state. In our training data sets, the maximum and minimum value of the electrical input power are 246.3W and 2608.6W, respectively.

B. Surrogate Models

We build surrogates for each output of the use cases. For the IM-M use case, the measured current ($i_{ps,re}$) is constant. The inputs and outputs of the different surrogates are presented in Figure 4.

All of the surrogate models are built with three different techniques. The first technique is the Gaussian Process (GP) regression with the Matérn $5/2$ kernel. The implementation of the GP surrogate models is carried out using the GPFlow library [18]. The other surrogate modelling techniques are Random Forest (RF) and one variant of NN, the Multi Layer Perception
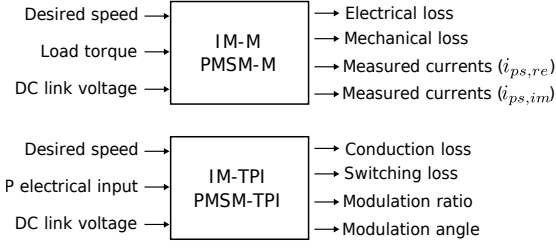
Fig. 4: The input-output behavior of the surrogate models.

(MLP). Both are implemented using the Scikit-learn library [19]. For the RF, 100 trees are used and no maximum depth is set. For MLP, we use 3 hidden layers with 150, 100 and 50 neurons, respectively. The Adam optimizer is used for learning the weights of the MLP for 500 iterations.

V. SIMULATION RESULTS AND VALIDATION

A. Model Validation and Testing

The performance of the surrogate models is evaluated using both 5-fold cross-validation [20] as well as independent test sets. A test set of 500 random points is generated for each model. The Mean Squared Error (MSE) is used as the performance metric.

The full results of the comparison are shown in Table II. We can see that the GP performs very well compared to the other surrogate models with a relatively low number of training data.

B. Time-domain Validation

A time-domain validation is performed by changing one of the value of the inputs continuously.

1) IM: In the IM-M scenario, we change the desired speed input linearly 1475rpm to 1375rpm starting from $t = 85$s to $t = 90$s. We consider surrogates for three outputs for this case: mechanical loss, electrical loss, and the stator current $i_{ps,im}$. The performance of the surrogate models is shown in Figure 5. The Mechanical loss surrogate model has a good prediction performance for the transient behavior. While, the electrical loss and the stator current $i_{ps,im}$ surrogates has a slightly biased prediction for the transient phase. Furthermore, after the transient period, the bias vanishes. The performance of the surrogates models in the IM-TPI use case under the same desired speed input changes is illustrated in Figure 6. There are no biases in the mean prediction of the overall phase, however in the start and the end of the transient phase, the conduction loss, switching loss, and modulation angle surrogates have a periodic behavior that our surrogates cannot capture.

2) PMSM: For the PMSM-M use case, we applied a change for the torque load input. The changes are applied from a steady-state for 12Nm to 5Nm, starting from $t = 85$s to $t = 90$s. The surrogates have a good performance as depicted in the Figure 7.

In the PMSM-TPI setting, we consider desired speed changes like the IM case, since we are not using torque load as an input. We can see from Figure 8 that the performance of the surrogate models are better than in the IM-TPI setting. The
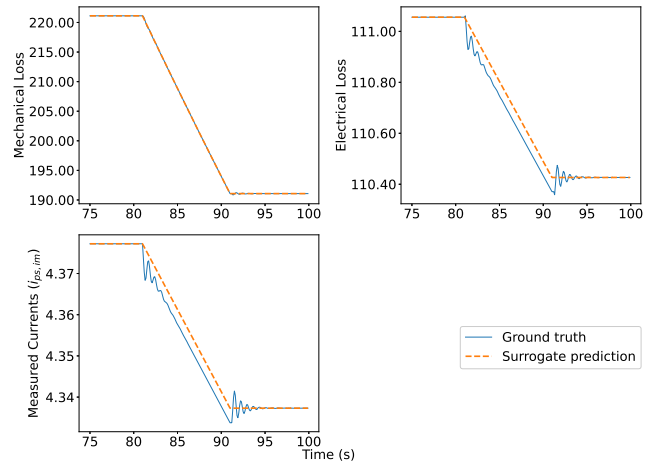

Fig. 5: IM surrogate model validation for transient behavior.
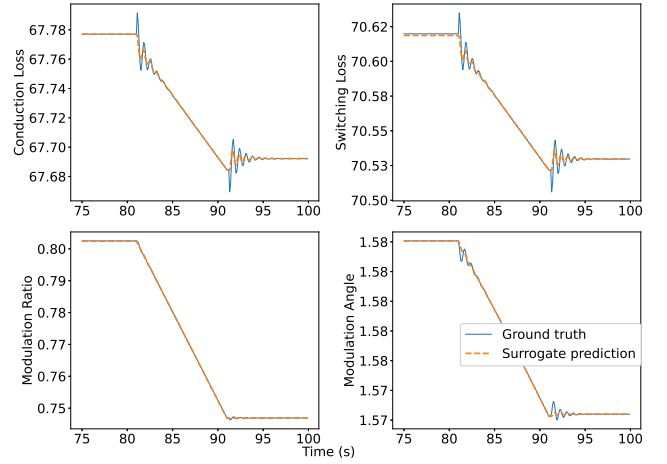

Fig. 6: TPI with IM transient behaviour validation under speed reference changes.

periodic behaviour at the start and the end of the transient phase has lower amplitude than the IM-TPI setting.
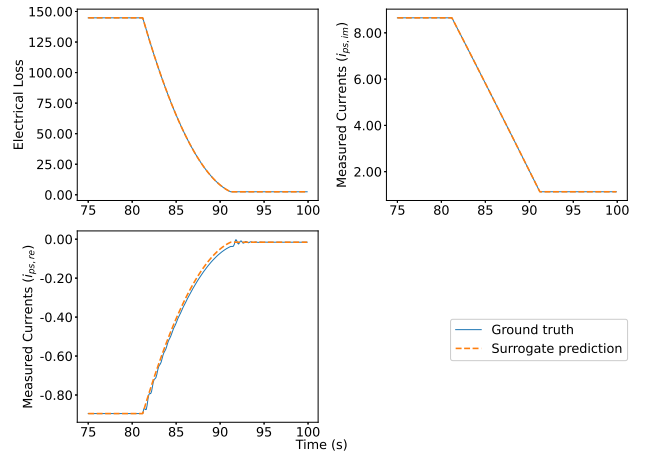

Fig. 7: PMSM surrogate models transient phase validation.

TABLE II: Validation and Testing Mean Squared Error (MSE) of all Surrogate Models

| Surrogate Model Name	Gaussian Process		Random Forest		Multi Layer Perceptron	
	K-Fold MSE	Test MSE	K-Fold MSE	Test MSE	K-Fold MSE	Test MSE
IM-M electrical loss	2.168e-5	**5.862e-6**	6.768e-1	4.085e-1	9.696e-1	5.654e-1
IM-M mechanical loss	1.820e-6	**7.192e-7**	1.177e-1	2.622e-2	1.548	7.029e-1
IM-M measured currents ($i_{ps,im}$)	8.659e-9	**1.883e-9**	3.451e-3	2.051e-3	3.327e-2	2.198e-3
PMSM-M electrical loss	2.160e-4	**1.654e-5**	1.128e+1	1.412	1.800e+1	7.537
PMSM-M mechanical loss	3.640e-6	**4.513e-7**	1.173e-1	6.180e-3	1.780	7.115e-1
PMSM-M measured currents ($i_{ps,re}$)	5.307e-9	**0.603e-9**	3.822e-4	5.164e-5	3.423e-3	2.306e-4
PMSM-M measured currents ($i_{ps,im}$)	1.443e-8	**2.703e-9**	1.324e-2	1.678e-3	8.352e-2	7.991e-3
IM-TPI conduction loss	6.646e-6	**7.390e-7**	7.354e-2	3.647e-2	1.945e-2	8.486e-3
IM-TPI switching loss	3.611e-4	**1.552e-5**	1.037	5.029e-1	1.445e-1	6.297e-2
IM-TPI modulation ratio	3.325e-9	**0.303e-9**	1.068e-4	4.276e-5	1.032e-5	6.279e-6
IM-TPI modulation angle	0.169e-3	**0.001e-9**	2.124e-7	1.346e-7	6.082e-7	7.930e-8
PMSM-TPI conduction loss	2.260e-4	**4.572e-6**	2.349	1.180	5.267	3.417e-1
PMSM-TPI switching loss	9.051e-4	**7.547e-4**	8.757	4.813	4.188	5.935e-1
PMSM-TPI modulation ratio	3.114e-9	**0.388e-9**	7.149e-5	3.005e-5	3.244e-6	1.768e-6
PMSM-TPI modulation angle	1.721e-8	**0.256e-9**	9.147e-5	4.918e-5	2.417	1.659e-5
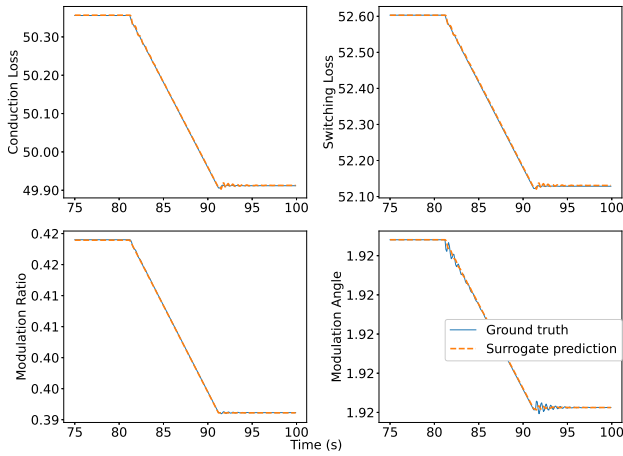
Fig. 8: TPI with PMSM surrogate models transient phase validation under speed reference changes.

VI. CONCLUSION AND FUTURE WORK

This paper benchmarks surrogate modelling techniques for steady-state dynamic phasor based IM and PMSM electrical drives. We conclude that GP surrogate models perform better than MLP and RF in terms of predictive performance. Also, GPs have desirable properties such as data efficiency and uncertainty estimation of their prediction. We also investigated the performance of the surrogates for predicting transient behavior, which shows promising performance.

Future work will focus on building transient surrogate models for explicitly modelling the transient behavior of the physical model. Further, the uncertainty quantification of the Gaussian Process could be utilized for various tasks such as active learning and anomaly detection. Finally, a more detailed speed comparison between the DP and the surrogate models in resource-constrained environments will be conducted.

REFERENCES

[1] Z. Xu, Y. Gao *et al.*, "Surrogate thermal model for power electronic modules using artificial neural network," in *IECON 2019-45th Annual Conference of the IEEE Industrial Electronics Society*, vol. 1. IEEE, 2019, pp. 3160–3165.

[2] P. L. Evans, A. Castellazzi *et al.*, "Automated fast extraction of compact thermal models for power electronic modules," *IEEE transactions on power electronics*, vol. 28, no. 10, pp. 4791–4802, 2012.

[3] T. Dragičević, P. Wheeler *et al.*, "Artificial intelligence aided automated design for reliability of power electronic systems," *IEEE Transactions on Power Electronics*, vol. 34, no. 8, pp. 7161–7171, 2018.

[4] G. E. Hinton, "Connectionist learning procedures," *Artificial Intelligence*, vol. 40, no. 1, pp. 185–234, 1989. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0004370289900490

[5] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. The MIT Press, 2018.

[6] J. Görtler, R. Kehlbeck *et al.*, "A visual exploration of gaussian processes," *Distill*, 2019, https://distill.pub/2019/visual-exploration-gaussian-processes.

[7] N. Quadrianto, K. Kersting *et al.*, *Gaussian Process*. Boston, MA: Springer US, 2010, pp. 428–439. [Online]. Available: https://doi.org/10.1007/978-0-387-30164-8_324

[8] L. Breiman, "Random forests," pp. 5–32, 2001.

[9] P. C. Krause, O. Wasynczuk *et al.*, *Analysis of electric machinery and drive systems*. John Wiley & Sons, 2013, vol. 75.

[10] J. D. Irwin, *The industrial electronics handbook*. CRC press, 1997.

[11] X. Ruan, X. Wang *et al.*, *Control techniques for LCL-type grid-connected inverters*. Springer, 2018.

[12] R. B. Gramacy, *Surrogates: Gaussian Process Modeling, Design and Optimization for the Applied Sciences*. Boca Raton, Florida: Chapman Hall/CRC, 2020.

[13] A. B. Owen, "A randomized halton algorithm in r," 6 2017. [Online]. Available: http://arxiv.org/abs/1706.02808

[14] R. H. Norden, "A survey of maximum likelihood estimation," *International Statistical Review / Revue Internationale de Statistique*, vol. 40, no. 3, pp. 329–354, 1972. [Online]. Available: http://www.jstor.org/stable/1402471

[15] B. Minasny and A. B. McBratney, "The Matérn function as a general model for soil variograms," *Geoderma*, vol. 128, no. 3-4 SPEC. ISS., pp. 192–207, 2005.

[16] M. Abadi, P. Barham *et al.*, "Tensorflow: A system for large-scale machine learning," in *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, 2016, pp. 265–283.

[17] A. Kalaiarasi Ganeson, P. Fritzon *et al.*, "An OpenModelica Python Interface and its use in PySimulator," *Proceedings of the 9th International MODELICA Conference, September 3-5, 2012, Munich, Germany*, vol. 76, no. August 2014, pp. 537–548, 2012.

[18] A. G. de G. Matthews, M. van der Wilk *et al.*, "Gpflow: A gaussian process library using tensorflow," *Journal of Machine Learning Research*, vol. 18, no. 40, pp. 1–6, 2017. [Online]. Available: http://jmlr.org/papers/v18/16-537.html

[19] F. Pedregosa, G. Varoquaux *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[20] M. Stone, "Cross-validatory choice and assessment of statistical predictions," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 36, no. 2, pp. 111–147, 1974. [Online]. Available: http://www.jstor.org/stable/2984809