Recursive Prediction Error Gradient-Based Algorithms and Framework to Identify PMSM Parameters Online

Aravinda Perera, *Student Member, IEEE,* Roy Nilsen

***Abstract*—Real-time acquisition of accurate machine parameters is of significance to achieving high performance in electric drives, particularly targeted for mission-critical applications. Unlike the saturation effects, the temperature variations are difficult to predict, thus it is essential to track temperature-dependent parameters online. In this paper, a unified framework is developed for online parameter identification of rotating electric machines, premised on the Recursive Prediction Error Method (RPEM). Secondly, the prediction gradient (Ψ^T)-based RPEM is adopted for identification of the temperature-sensitive parameters, i.e., the permanent magnet flux linkage (Ψ_m) and stator-winding resistance (R_s) of the Interior Permanent Magnet Synchronous Machine (IPMSM). Three algorithms, namely, Stochastic Gradient (SGA), Gauss-Newton (GNA), and physically interpretative method (PhyInt) are investigated for the estimation gains computation. A speed-dependent gain-scheduling scheme is used to decouple the inter-dependency of Ψ_m and R_s. With the aid of offline simulation methods, the main elements of RPEM such as Ψ^T are analyzed. The concept validation and the choice of the optimal algorithm is made with the use of System-on-Chip (SoC) based Embedded Real-Time Simulator (ERTS). Subsequently, the selected algorithms are validated with the aid of a 3-kW, IPMSM drive where the control and estimation routines are implemented in the SoC-based industrial embedded control system. The experimental results reveal that Ψ^T-based RPEM, in general, can be a versatile technique in temperature-sensitive parameter adaptation both online and offline.**

***Index Terms*—Gain-matrix, gain-scheduling, Gauss-Newton, PMSM, prediction-error, stochastic gradient, variable speed drive**

I. Introduction

At the wake of electrification in the operational reliability and safety-critical applications such as surface transport, aerospace, and seabed mining, the dependability of the electrical systems becomes of major significance. Also, the increasing urge to reduce the carbon footprint calls upon more efficient power systems. IPMSM-equipped electric drives become a frontrunner in this context, owing to some of their inherent features such as superior efficiency and power density, thus ease of cooling, design capability for fault-tolerance, and good control dynamics in a wide torque-speed range [1].

In realizing a high-performance electric drive, the knowledge of exact machine parameters is essential for multiple reasons [2], yet the exact parameters are often unknown across the operating range. It is, therefore, useful to identify the machine parameters of the electric drive, thus a variety of online and offline identification methods as reviewed in [2], [3] have gained attention in recent years. Out of the electric parameters, i.e. Ψ_m, R_s and d- and q- axis inductances L_d, L_q, the first two are temperature-dependent, and display slow dynamics due to the thermal capacity. L_d and L_q can vary rapidly as they are iron core-saturation dependent, a phenomenon that is dictated by the stator current. Nevertheless, simultaneous identification of more than two unknown parameters is prohibited by the rank-deficiency problem of IPMSM [4] unless extra efforts are exerted. Adoption of two time-scale routines for fast- and slow- dynamic parameter-sets [5] or High-Frequency Signal Injection (HFSI) [6] or a combination of such methods [7] or the use of special instrumentation for online computation of inductances as reported in [8] have been employed to solve the rank-deficiency challenge. Alternatively, some of the electric parameters can be identified offline and updated online. Stator current which affects the inductances, is a measured quantity in electric drives, thus, the model inductances can be updated with reference to stator current in real-time, using the offline identified inductances with the aid of a Look-Up Table (LUT) or analytical functions [9]. Sensor-based R_s and Ψ_m acquisition have also been reported both with and without contact [10], [11]. Sensor-based stator temperature monitoring is a more common industrial approach, from which R_s and Ψ_m can be back-calculated in real-time. Although, such sensor-based acquisitions are associated with considerable integration- and reliability- concerns and also will have implications in the space requirement and the overall costs [12]. Online tracking of R_s and Ψ_m can be a more versatile solution from this perspective. Based on this premise of the two fold benefits, R_s and Ψ_m are identified online and the inductances are identified offline and updated online, in the scope of this paper, which will enable us to solve the rank deficiency problem in addition to drive performance enhancement.

A. Literature Review

The RPEM is a set of parameter identification methods presented by Ljung [13], in which it is indicated that several well-known techniques like the Recursive Least Squares (RLS) and the Extended Kalman Filter (EKF) methods can be viewed as its subsets. Among these, RLS, perhaps the most widely adopted, is used in [5] for the identification of all electric parameters of PMSM. It is reported in [14], the use of RLS to improve the performance of the Model Predictive Controlled PMSM by recursively updating the prediction models. In sensorless drives, the position-estimation accuracy is enhanced using RLS in [15], [16]. A combination of a signal injection scheme and the RLS method is applied in [7] to identify IPMSM parameters

of a Direct Torque Control drive. The EKF, another popular member of the RPEM family is discussed for online parameter adaptation [17], [18] offers decent performance at the cost of increased computational burden. An alternative method under RPEM-family, that exploits the sensitivity of the predicted currents to the model parameters has been discussed in [13], in which, this method is termed as prediction gradient ($\boldsymbol{\Psi}^T$)-based Recursive Prediction Error Method (RPEM). $\boldsymbol{\Psi}^T$-based RPEM offers more consistent estimations [19] and the global convergence is more often guaranteed [20], [21] compared to EKF-based identification. Additionally, opposing to RLS or EKF methods, the digital implementation of $\boldsymbol{\Psi}^T$-based RPEM can be less demanding due to the possibility of avoiding the tedious computations like the matrix inversions. Model Reference Adaptive System (MRAS)-based parameter identification has been adopted in [22], [23]. Apart from its known stability issues and design complexities [2], the reference model is assumed to emulate the plant, which may not be the case if its parameters mismatch the physical counterparts, thus the prediction error may not represent the true parameter discrepancy. A variety of artificial intelligence applications like different forms of neural network (NN) models and evolutionary algorithms like the Particle Swamp Optimization (PSO) have been reported in connection to the parameter estimation of PMSM [24], [25]. To overcome their known -stability and -convergence issues [26], more advance methods, for example, with self-learning capability, have emerged lately, however with the penalty of longer model-training time or demand of high expertise in design and implementation [26].

B. Research Gaps and Contribution

In spite of the merits of $\boldsymbol{\Psi}^T$-based RPEM, it has not been investigated in the last decades, thus omitted in the recent reviews [2], [3]. Another notable research gap in the RPEM-related literature is the absence of basis and underlying principles behind the choices of estimation gains. This article attempts, firstly, to adopt the $\boldsymbol{\Psi}^T$-based RPEM for online parameter identification of IPMSM, an investigation that has not been done before, to the authors' best knowledge. Three algorithms, namely SGA, GNA and PhyInt become applicable under this context [13], [27]. The SGA and GNA -based Ψ_m and R_s identification using the offline simulation tools, presented in [28] will be extended with the real-time simulation and experimental validation in this article. Additionally, PhyInt is also explored for Ψ_m and R_s identification. Eventually, the performances with different algorithms are compared to draw conclusions for optimal algorithm to compute estimation-gains for $\boldsymbol{\Psi}^T$-based RPEM. Secondly, to fill the absence of an elaborate procedure to identify estimation-gains in the drives domain, a general approach outlined in [13] is tailored for electric drives with the aim of formulating a thorough and physically insightful framework for RPEM-based identification. The step-by-step sequence explicitly: 1) Choice of Model-Set, $\mathscr{M}$; 2) Choice of experimental conditions; 3) Choice of criterion function; 4) Choice of search direction using prediction gradient; 5) Choice of gain-sequence and initial values. In order to focus the scope to parameter identification, a mechanical position-sensor is assumed to obtain the rotor position although the incorporation of position-sensorless schemes within the same scope is possible. A Zynq System-on-Chip (SoC) based ERTS and a 3 kW-IPMSM experimental setup is used for simulation and experimental validation.

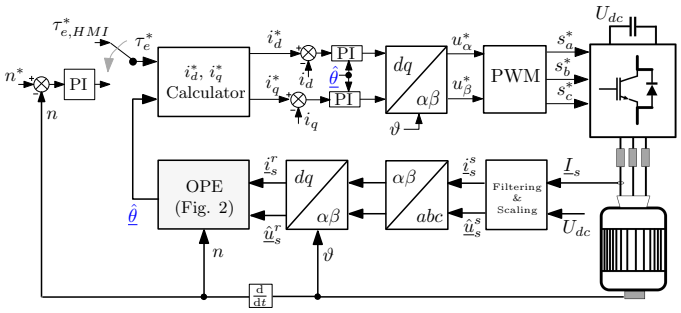


Fig. 1: Block Diagram of the Field Oriented Controlled IPMSM Drive enhanced with the Online Parameter Estimator.

C. Organization

The paper is organized in the following manner. In Section II, the IPMSM model and control is briefly outlined. The proposed framework and explicit development of $\boldsymbol{\Psi}^T$-based RPEM is unfolded in the Section III, where the above mentioned sequence is followed. Section IV explores the use of a rotor-speed dependent gain-scheduler to circumvent the cross-coupling effects between Ψ_m and R_s. Subsequently, the validation results and discussions are revealed using the ERTS in the Section V and using the experimental setup in the Section VI, while the concluding remarks are contained in the Section VII.

II. IPMSM Modeling and Control

In this section, the dynamic model of the IPMSM and its Field Oriented Control (FOC) is outlined. The mathematical model of the electrical part of the machine is in the rotor coordinates when given in the per-unit (pu) system as per [29]:

$$\begin{aligned} \underline{u}^r_s &= r_s \cdot \underline{i}^r_s + \frac{1}{\omega_n} \cdot \frac{\mathrm{d}\underline{\psi}^r_s}{\mathrm{d}t} + \mathbf{j} \cdot n \cdot \underline{\psi}^r_s \\ \underline{\psi}^r_s &= \mathbf{x}^r_s \cdot \underline{i}^r_s + \underline{\psi}^r_m \end{aligned} \tag{1}$$

$$\underline{i}^r_s = \begin{bmatrix} i_d & i_q \end{bmatrix}^T, \quad \underline{\psi}^r_m = \begin{bmatrix} \psi_m & 0 \end{bmatrix}^T$$
$$\mathbf{x}^r_s = \begin{bmatrix} x_d & 0 \\ 0 & x_q \end{bmatrix}, \quad \mathbf{j} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

Here, $u, i, \psi, x, n, \omega_n$ are voltage, current, flux linkage, inductances, electric speed, and nominal rotational frequency respectively. ϑ is the electrical angle of the mechanical position ϑ_{mech} whose relationship with ϑ is given by $\vartheta = p \cdot \vartheta_{mech}$ where p is the number of pole pairs. Throughout the article, the superscript and subscript denote the reference frame and the location of the quantity (s-stator, r-rotor, m-magnet) respectively. The notation $\hat{}$ and superscript * indicate the estimated and the reference -quantities respectively.

The block diagram of the three-phase, FOC, IPMSM drive enhanced by the Online Parameter Estimator (OPE) is given in Fig. 1. The classical two-level, three-phase, Insulated-Gate Bipolar Transistor (IGBT), Voltage Source Inverter (VSI) is

supplied by dc-link capacitors, in which the voltage U_{dc} is measured and used to estimate the stator winding voltages while compensating for the dead-time and the device on-stage voltage drop as given in [30]. $\underline{I}_s$ is measured at the output of the VSI. The OPE estimates the model parameter vector $\hat{\underline{\theta}}$ that is fed into the reference calculator and Proportional-Integral (PI) controllers. Based on the given torque command τ_e^* either from the Human Machine Interface (HMI) or from the speed controller, i_d^*, i_q^* are calculated to fulfill either MTPA using or the field-weakening strategy at high-speed operations as described in [31].

III. Proposed Framework and Development of $\boldsymbol{\Psi}^T$-based RPEM

To begin with, RPEM can be generalized as in (2).

$$\hat{\underline{\theta}}[k] = \left[\hat{\underline{\theta}}[k-1] + \mathbf{L}[k,\hat{\underline{\theta}}] \cdot \underline{\epsilon}[k,\hat{\underline{\theta}}]\right]_{D\mathscr{M}} \tag{2}$$

Here, $\mathbf{L}$ is the gain-matrix, $\underline{\epsilon}$ is the criterion function that we attempt to minimize and eventually nullify, choosing appropriate $\mathbf{L}$. Among the various approaches to compute $\mathbf{L}$, we adopt $\boldsymbol{\Psi}^T$-based methods. Opposing to the common practice, in this section, we aim to reveal the underlying principles of computing $\mathbf{L}$ by adopting the step-by-step approach [13] from, for online identification of three-phase IPMSM parameters.

A. Choice of Model-Set, $\mathscr{M}(\underline{\theta})$

The Full-Order Model, $\mathscr{M}_{u\theta}$, given by (3) is chosen under the proposed method because it incorporates the electric parameters of interest. $\mathscr{M}_{u\theta}$ is used to construct a predictor to predict the stator current, $\hat{\underline{i}}_s^r$. The prediction-error, $\underline{\epsilon}_s^r$ is then generated using the measured and the predicted currents, which can be expressed in discrete form as $\underline{\epsilon}_s^r = \underline{i}_s^r[k] - \hat{\underline{i}}_s^r[k,\hat{\underline{\theta}}]$. It is assumed that the sole cause for nonzero $\underline{\epsilon}_s^r$ is the difference between the physical and model parameters. The block diagram of the $\mathscr{M}_{u\theta}$-based OPE is given in the Fig. 2. $\underline{\epsilon}_s^r$ is fed forward instead of feedback correction mechanism, unlike in a closed-loop/observer structure. Therefore, this open-loop predictor arrangement enriches $\underline{\epsilon}_s^r$ with parameteric error information, a feature that is attempted to capitalize in computing the prediction gradients under this method. $\underline{\epsilon}_s^r$ is discussed in detail in the Section III-C. This choice, however, requires a separate Model-Set for parameter estimation, unlike the MRAS-based approaches [22], [23], in which both parameter- and state- estimations can be integrated within one Model-Set. Thus, when low-cost processor-based or high-speed drives are concerned, the designers should be cautious in the additional time taken by the proposed scheme in the processor.

$$\begin{aligned}\frac{\hat{\mathbf{x}}_s^r(\underline{i}_s)}{\omega_n}\frac{\mathrm{d}\hat{\underline{i}}_s^r(t,\hat{\underline{\theta}})}{\mathrm{d}t} &= \underline{u}_s^r(t) - \hat{r}_s(t)\cdot\hat{\underline{i}}_s^r(t,\hat{\underline{\theta}}) \\ &\quad - \mathbf{j}\cdot n(t)\cdot\hat{\mathbf{x}}_s^r(\underline{i}_s)\cdot\hat{\underline{i}}_s^r(t,\hat{\underline{\theta}}) - \mathbf{j}\cdot n(t)\cdot\underline{\psi}_m^r(t) \\ \underline{i}_s^r &= \mathbf{T}_{ss}^r(\vartheta)\cdot\underline{i}_s^s, \qquad \underline{u}_s^r = \mathbf{T}_{ss}^r(\vartheta)\cdot\underline{u}_s^s\end{aligned} \tag{3}$$

$\mathscr{M}_{u\theta}$ is a second-order system in which the linearized system matrix $\mathbf{A}$ and the eigenvalues, $\lambda_{1,2}$ are given in the (4a) and (4b) respectively, where $\hat{T}_d, \hat{T}_q$, expressed in (4c) are d,q- axes time-constants. In order to linearize the nonlinear IPMSM model, we consider only the electrical equations, in which, n is treated as a measured and slow-varying parameter. Fig. 3(a) plots the trajectories of $\lambda_{1,2}$ against the increasing rotor speed from standstill for the IPMSM given in the Table I. It is evident that the $\mathscr{M}_{u\theta}$ is stable across the full speed range yet, $\hat{\underline{i}}_s^r$ can contain oscillations and their frequency is expected to increase in proportion to the rotor speed. Due to this speed-dependency, the numerical method adopted to discretize $\mathscr{M}_{u\theta}$ as well as the integration time-step of the digital controller can influence the stability of the digitally implemented predictor. Unlike the explicit Euler method, the trapezoidal rule based numerical method has much larger stability region in the $\lambda - T_{samp}$-plane [32], that can guarantee the full speed-range stability of $\mathscr{M}_{u\theta}$-based open-loop predictor when implemented in a processor at sampling times T_{samp} corresponding to IGBT-based drives. Fig. 3(b) illustrates, how eigenvalues escape the Euler-based stability region in $\lambda - T_{samp}$-plane, yet are well within that of the trapezoidal rule, when $T_{samp} = 125\,\mu s$. Eq. (5) expresses the predicted currents when discretized using the trapezoidal rule-based numerical method, in which, $\hat{\underline{i}}_s^r[k]$ must be explicitly solved w.r.t. $\hat{\underline{i}}_s^r[k-1]$ and other variables.

$$\lambda\cdot I_2 - A = \begin{bmatrix} \lambda + \frac{1}{\hat{T}_d} & \frac{-n\cdot x_q\cdot\omega_n}{x_d} \\ \frac{-n\cdot x_d\cdot\omega_n}{x_q} & \lambda + \frac{1}{\hat{T}_q} \end{bmatrix} \tag{4a}$$

$$\lambda_{1,2} = -\frac{1}{2}\cdot\left(\frac{1}{\hat{T}_d} + \frac{1}{\hat{T}_q}\right) \pm \sqrt{\left[\frac{1}{2}\cdot\left(\frac{1}{\hat{T}_d} + \frac{1}{\hat{T}_q}\right)\right]^2 - \left[\left(\frac{1}{\hat{T}_d\cdot\hat{T}_q}\right) + (\omega_n\cdot n)^2\right]} \tag{4b}$$

$$\hat{T}_d = \frac{x_d}{\hat{r}_s\cdot\omega_n}, \qquad \hat{T}_q = \frac{x_q}{\hat{r}_s\cdot\omega_n} \tag{4c}$$

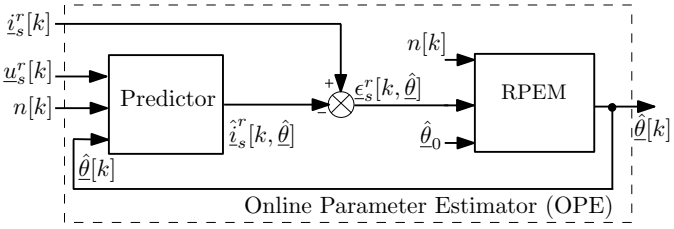
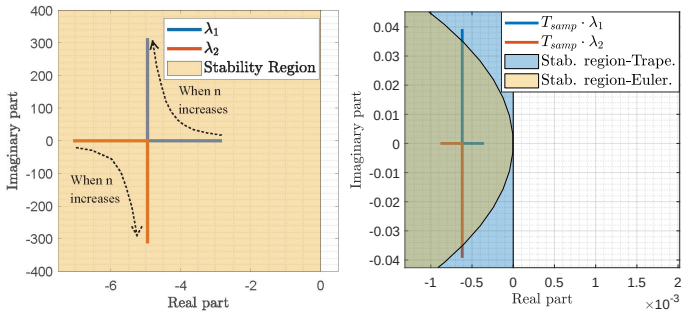


Fig. 2: Block diagram of the Open-Loop Online Parameter Estimator.



(a) (b)

Fig. 3: Eigenvalue trajectories and stability regions (a) in continuous time domain (b) when discretized using trapezoidal method, in $\lambda - T_{samp}$ plane.

$$\hat{\underline{i}}_s^r[k,\hat{\underline{\theta}}] = \hat{\underline{i}}_s^r[k-1,\hat{\underline{\theta}}] + \frac{T_{samp}}{2}\left(\underline{g}_c\left(\hat{\underline{i}}_s^r[k,\hat{\underline{\theta}}], \underline{u}_s^r[k], n[k]\right) + \underline{g}_c\left(\hat{\underline{i}}_s^r[k-1,\hat{\underline{\theta}}], \underline{u}_s^r[k-1], n[k-1]\right)\right) \quad (5)$$

B. Choice of Experimental Conditions

The choice of experimental conditions imply *when* and *which* data are collected from the process for the identification. The input signals for the OPE model are identified as $\underline{u}_s^r, \underline{i}_s^r, n$ as illustrated in the Fig. 2. An online identification method, both at the start-of the drive and during its operation, is chosen as the means to acquire the input signals in order to identify Ψ_m and R_s. Hence, $\hat{\underline{\theta}}$ becomes as in (6a). In [13] it is shown that to guarantee global convergence, the $\hat{\underline{\theta}}$ must be bounded by the parameter-space D_s, that defines the stable region of the $\mathscr{M}_{u\theta}$-based predictor. To facilitate faster tracking, a narrower parameter-subspace, $D_{\mathscr{M}}$ can be defined as given in the (6b).

In addition to the input signals to the OPE, ϑ needs to be accurately identified, because of the required reference frame transformations in the OPE and FOC in general.

$$\hat{\underline{\theta}} = \begin{bmatrix}\hat{\psi}_m & \hat{r}_s\end{bmatrix}^T, \qquad \hat{\underline{\theta}} \in D_{\mathscr{M}}, \qquad D_{\mathscr{M}} \in D_s \quad (6a)$$

$$D_{\mathscr{M}} = \left\{\begin{matrix}\hat{\psi}_{m,min} \le \hat{\psi}_m \le \hat{\psi}_{m,max} \\ \hat{r}_{s,min} \le \hat{r}_s \le \hat{r}_{s,max}\end{matrix}\right\} \quad (6b)$$

C. Choice of Criterion Function, $V_N(\hat{\underline{\theta}})$

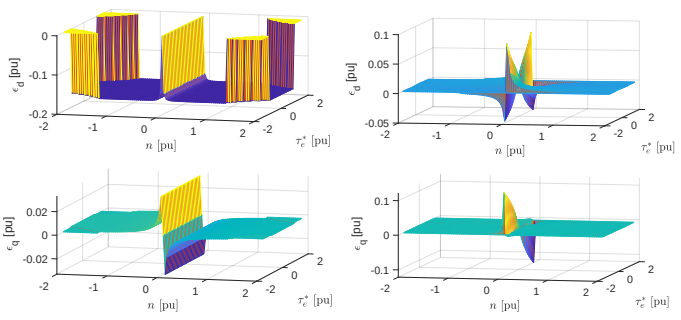
$V_N(\hat{\underline{\theta}})$ and its asymptotic properties are influenced by the choice of $\mathscr{M}(\underline{\theta})$. If a Gaussian distribution of the prediction errors is assumed, $V_N(\hat{\underline{\theta}})$ becomes a scalar quadratic criterion [13] as given in (7) in which Λ is the covariance matrix of the prediction error.

$$V_N(\hat{\underline{\theta}}) = \frac{1}{2}\underline{\epsilon}_s^{rT}(t,\hat{\underline{\theta}}) \cdot \Lambda^{-1} \cdot \underline{\epsilon}_s^r(t,\hat{\underline{\theta}}) \quad (7)$$

Assuming that the Λ is known and independent of model-parameters, and the prediction error is based on the current measurement, Λ is chosen as the Identity Matrix.

The sensitivity of the prediction error to all four parametric errors can be evaluated by deriving an expression for the steady-state $\underline{\epsilon}_s^r$ in component form as in (8).

$$\begin{aligned}\epsilon_d = &-\left(\frac{n^2\cdot\hat{x}_q}{\hat{r}_s^2+n^2\cdot\hat{x}_d\cdot\hat{x}_q}\right)\delta\psi_m \\ &-\left(\frac{\hat{r}_s}{\hat{r}_s^2+n^2\cdot\hat{x}_d\cdot\hat{x}_q}\cdot i_d + \frac{n\cdot\hat{x}_q}{\hat{r}_s^2+n^2\cdot\hat{x}_d\cdot\hat{x}_q}\cdot i_q\right)\delta r_s \\ &-\left(\frac{n^2\hat{x}_q}{\hat{r}_s^2+n^2\cdot\hat{x}_d\cdot\hat{x}_q}\cdot i_d\right)\delta x_d \\ &+\left(\frac{n\cdot\hat{r}_s}{\hat{r}_s^2+n^2\cdot\hat{x}_d\cdot\hat{x}_q}\cdot i_q\right)\delta x_q\end{aligned}$$



(a) (b)

Fig. 4: Prediction-errors in 4-quadrant speed-torque plane when estimate is 10% lower than respective physical quantity w.r.t. (a) $\hat{\psi}_m$ (b) $\hat{r}_s$.

$$\begin{aligned}\epsilon_q = &-\left(\frac{n\cdot\hat{r}_s}{\hat{r}_s^2+n^2\cdot\hat{x}_d\cdot\hat{x}_q}\right)\delta\psi_m \\ &-\left(\frac{\hat{r}_s}{\hat{r}_s^2+n^2\cdot\hat{x}_d\cdot\hat{x}_q}\cdot i_q - \frac{n\cdot\hat{x}_d}{\hat{r}_s^2+n^2\cdot\hat{x}_d\cdot\hat{x}_q}\cdot i_d\right)\delta r_s \\ &-\left(\frac{n\cdot\hat{r}_s}{\hat{r}_s^2+n^2\cdot\hat{x}_d\cdot\hat{x}_q}\cdot i_d\right)\delta x_d \\ &-\left(\frac{n^2\cdot\hat{x}_d}{\hat{r}_s^2+n^2\cdot\hat{x}_d\cdot\hat{x}_q}\cdot i_q\right)\delta x_q\end{aligned}$$

$$\delta\psi_m = \psi_m - \hat{\psi}_m, \qquad \delta r_s = r_s - \hat{r}_s$$
$$\delta x_d = x_d - \hat{x}_d, \qquad \delta x_q = x_q - \hat{x}_q \quad (8)$$

To remain within the scope of the article, let us assume the model inductances are in alignment with their physical counterparts, thus $\delta x_d, \delta x_q = 0$ in (8). Therein, the prediction error sensitivities can be visualized in the 4-quadrant speed-torque plane w.r.t. a 10% underestimation in $\hat{\psi}_m$ in Fig. 4 (a) and a 10% underestimation in $\hat{r}_s$ in 4 (b). In connection to (8) and Fig. 4, the following observations can be remarked.

Remark 1: When $\delta\psi_m$, δr_s becomes zero, $\epsilon_{d,q}$ also go to zero.

Remark 2: When $\delta\psi_m$ is concerned (see Fig. 4 (a)) , ϵ_d is consistently well-condition with $\delta\psi_m$ beyond very low rotor speeds. When n increases, $\epsilon_d \approx \frac{-1}{\hat{x}_d}\cdot\delta\psi_m$. On the contrary, the sensitivity of ϵ_q to $\delta\psi_m$ across the operating range is weak and inconsistent to make ϵ_q redundant information for ψ_m-identification.

Remark 3: When δr_s is concerned (see Fig. 4 (b)), both ϵ_d and ϵ_q become dominant at and around zero-speed to carry rich-conditioned information for r_s-identification.

Remark 4: When δr_s is concerned, $\epsilon_{d,q}$ are also stator current dependent, meaning, even at standstill, $\underline{\epsilon}_s^r$ carries information to identify r_s if stator current is present.

Remark 5: $\epsilon_{d,q}$ becomes more sensitive to $\delta\psi_m$ and δr_s in mutually exclusive speed regions. The dominance of δr_s-sensitivity is at and around zero speed and this is the very region, the accuracy of $\hat{r}_s$ becomes critical when the Voltage Model based computations are concerned.

D. Choice of Search Direction using Prediction Gradient, Ψ^T

Once V_N is chosen, the correct direction to minimize V_N is discovered using a search direction algorithm. In this article, we

focus on algorithms that rely on $\boldsymbol{\Psi}^T$, which will be developed in this section.

One well-known numerical minimization approach is the use of gradient of the criterion function. It is shown in [13] that in the pursuit of ∇V, the prediction-error gradient, $\frac{\mathrm{d}\underline{\epsilon}_s^r}{\mathrm{d}\hat{\underline{\theta}}}$ becomes the actual gradient of interest. The prediction-error gradient becomes the negative of the prediction gradient, $\boldsymbol{\Psi}^T$ as been deduced in (9)

$$\begin{aligned}\frac{\mathrm{d}\underline{\epsilon}_s^r[k,\hat{\underline{\theta}}]}{\mathrm{d}\hat{\underline{\theta}}} &= \frac{\mathrm{d}\underline{i}_s^r[k,\vartheta]}{\mathrm{d}\hat{\underline{\theta}}} - \frac{\mathrm{d}\hat{\underline{i}}_s^r[k,\hat{\underline{\theta}}]}{\mathrm{d}\hat{\underline{\theta}}} \\ \frac{\mathrm{d}\underline{\epsilon}_s^r[k,\hat{\underline{\theta}}]}{\mathrm{d}\hat{\underline{\theta}}} &= -\frac{\mathrm{d}\hat{\underline{i}}_s^r[k,\hat{\underline{\theta}}]}{\mathrm{d}\hat{\underline{\theta}}} = -\boldsymbol{\Psi}^T[k,\hat{\underline{\theta}}] \end{aligned} \tag{9}$$

The dynamic forms of the $\boldsymbol{\Psi}^T$ can be derived by derivation of (3) w.r.t. $\hat{\psi}_m$ and $\hat{r}_s$. Such derivation can be generalized as in (10), in which, f_c is a function equivalent to the right hand side of (3). After solving, one can arrive at the respective gradient functions as shown in (11) and (12).

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}\hat{\underline{\theta}}}\left(\frac{\mathrm{d}\hat{\underline{i}}_s^r(t,\hat{\underline{\theta}}))}{\mathrm{d}t}\right) &= \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\mathrm{d}\hat{\underline{i}}_s^r(t,\hat{\underline{\theta}}))}{\mathrm{d}\hat{\underline{\theta}}}\right) \\ &= \frac{\mathrm{d}}{\mathrm{d}\hat{\underline{\theta}}}\left(\underline{f}_c\left(t,\hat{\underline{i}}_s^r(t,\hat{\underline{\theta}}),\underline{u}_s^r(t),n(t);\hat{\underline{\theta}}\right)\right)\end{aligned} \tag{10}$$

$$\frac{\mathrm{d}\left(\frac{\mathrm{d}\hat{i}_d}{\mathrm{d}\hat{\psi}_m}\right)}{\mathrm{d}t} = \frac{\omega_n}{\hat{x}_d}\left(-\hat{r}_s\cdot\frac{\mathrm{d}\hat{i}_d}{\mathrm{d}\hat{\psi}_m} + \hat{x}_q\cdot n\cdot\frac{\mathrm{d}\hat{i}_q}{\mathrm{d}\hat{\psi}_m}\right) \tag{11a}$$

$$\frac{\mathrm{d}\left(\frac{\mathrm{d}\hat{i}_q}{\mathrm{d}\hat{\psi}_m}\right)}{\mathrm{d}t} = \frac{\omega_n}{\hat{x}_q}\left(-\hat{r}_s\cdot\frac{\mathrm{d}\hat{i}_q}{\mathrm{d}\hat{\psi}_m} - \hat{x}_d\cdot n\cdot\frac{\mathrm{d}\hat{i}_d}{\mathrm{d}\hat{\psi}_m} - n\right) \tag{11b}$$

$$\frac{\mathrm{d}\left(\frac{\mathrm{d}\hat{i}_d}{\mathrm{d}\hat{r}_s}\right)}{\mathrm{d}t} = \frac{\omega_n}{\hat{x}_d}\left(-\hat{r}_s\cdot\frac{\mathrm{d}\hat{i}_d}{\mathrm{d}\hat{r}_s} + n\cdot\hat{x}_q\cdot\frac{\mathrm{d}\hat{i}_q}{\mathrm{d}\hat{r}_s} - \hat{i}_d\right) \tag{12a}$$

$$\frac{\mathrm{d}\left(\frac{\mathrm{d}\hat{i}_q}{\mathrm{d}\hat{r}_s}\right)}{\mathrm{d}t} = \frac{\omega_n}{\hat{x}_q}\left(-\hat{r}_s\cdot\frac{\mathrm{d}\hat{i}_q}{\mathrm{d}\hat{r}_s} - \hat{x}_d\cdot n\cdot\frac{\mathrm{d}\hat{i}_d}{\mathrm{d}\hat{r}_s} - \hat{i}_q\right) \tag{12b}$$

For the sake of completeness, the dynamic forms of the prediction gradients w.r.t. $\hat{x}_d, \hat{x}_q$ can be presented as follows, despite they are not in use under this scope of work.

$$\begin{aligned}\frac{\mathrm{d}\left(\frac{\mathrm{d}\hat{i}_d}{\mathrm{d}\hat{x}_d}\right)}{\mathrm{d}t} = \frac{\omega_n}{\hat{x}_d}\Bigg(&-\hat{r}_s\cdot\frac{\mathrm{d}\hat{i}_d}{\mathrm{d}\hat{x}_d} + \hat{x}_q\cdot n\cdot\frac{\mathrm{d}\hat{i}_q}{\mathrm{d}\hat{x}_d} + \frac{\hat{r}_s}{\hat{x}_d}\cdot\hat{i}_d \\ &-\frac{\hat{x}_q\cdot n}{\hat{x}_d}\cdot\hat{i}_q - \frac{u_d}{\hat{x}_d}\Bigg)\end{aligned} \tag{13a}$$

$$\frac{\mathrm{d}\left(\frac{\mathrm{d}\hat{i}_q}{\mathrm{d}\hat{x}_d}\right)}{\mathrm{d}t} = \frac{\omega_n}{\hat{x}_q}\left(-\hat{r}_s\cdot\frac{\mathrm{d}\hat{i}_q}{\mathrm{d}\hat{x}_d} - \hat{x}_d\cdot n\cdot\frac{\mathrm{d}\hat{i}_d}{\mathrm{d}\hat{x}_d} - n\cdot\hat{i}_d\right) \tag{13b}$$

$$\frac{\mathrm{d}\left(\frac{\mathrm{d}\hat{i}_d}{\mathrm{d}\hat{x}_q}\right)}{\mathrm{d}t} = \frac{\omega_n}{\hat{x}_d}\left(-\hat{r}_s\cdot\frac{\mathrm{d}\hat{i}_d}{\mathrm{d}\hat{x}_q} + n\cdot\hat{x}_q\cdot\frac{\mathrm{d}\hat{i}_q}{\mathrm{d}\hat{x}_q} + n\cdot\hat{i}_q\right) \tag{14a}$$

$$\begin{aligned}\frac{\mathrm{d}\left(\frac{\mathrm{d}\hat{i}_q}{\mathrm{d}\hat{x}_q}\right)}{\mathrm{d}t} = \frac{\omega_n}{\hat{x}_q}\Bigg(&-\hat{r}_s\cdot\frac{\mathrm{d}\hat{i}_q}{\mathrm{d}\hat{x}_q} - \hat{x}_d\cdot n\cdot\frac{\mathrm{d}\hat{i}_d}{\mathrm{d}\hat{x}_q} + \frac{\hat{r}_s}{\hat{x}_q}\cdot\hat{i}_q \\ &-\frac{\hat{x}_d\cdot n}{\hat{x}_q}\cdot\hat{i}_d - \frac{u_q}{\hat{x}_q}\Bigg)\end{aligned} \tag{14b}$$

The above dynamic forms of $\boldsymbol{\Psi}^T$ share the same eigenvalues with $\mathscr{M}_{u\theta}$, thus the concerns regarding the digital implementation discussed in the section III-A apply to these as well. The corresponding steady-state $\boldsymbol{\Psi}^T$ forms can be derived by equalizing the the left hand side of the each of the above equations to zero. The final derivations are given in the (15) and (16) w.r.t. $\hat{\psi}_m$ and $\hat{r}_s$, which can, in fact, be obtained by partially deriving (8) w.r.t. each parameter estimate. The steady-state $\boldsymbol{\Psi}^T$-functions are plotted in the Fig. 5.

$$\begin{aligned}\frac{\mathrm{d}\hat{i}_d}{\mathrm{d}\hat{\psi}_m} &= -\frac{n^2\cdot x_q}{\hat{r}_s + n^2\cdot x_q\cdot x_d},\ \frac{\mathrm{d}\hat{i}_q}{\mathrm{d}\hat{\psi}_m} = -\frac{n\cdot\hat{r}_s}{\hat{r}_s + n^2\cdot x_q\cdot x_d} \\ \frac{\mathrm{d}\hat{i}_d}{\mathrm{d}\hat{r}_s} &= -\frac{\hat{r}_s\cdot\hat{i}_d}{\hat{r}_s + n^2\cdot x_q\cdot x_d} - \frac{n\cdot x_q\cdot\hat{i}_q}{\hat{r}_s + n^2\cdot x_q\cdot x_d} \\ \frac{\mathrm{d}\hat{i}_q}{\mathrm{d}\hat{r}_s} &= -\frac{\hat{r}_s\cdot\hat{i}_q}{\hat{r}_s + n^2\cdot x_q\cdot x_d} + \frac{n\cdot x_d\cdot\hat{i}_d}{\hat{r}_s + n^2\cdot x_q\cdot x_d}\end{aligned} \tag{15, 16}$$

Based on the steady-state functions and corresponding plots, the following remarks can be made.

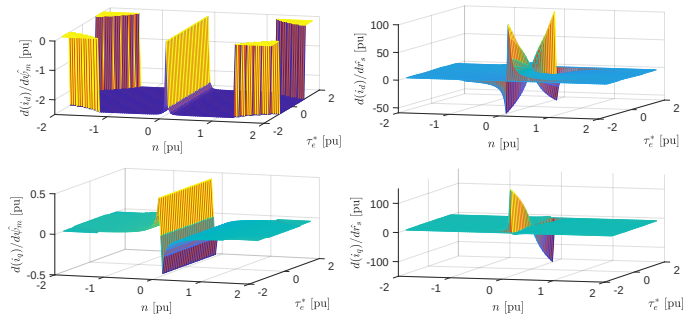
Remark 1: $\boldsymbol{\Psi}^T$-steady state forms hold the same shapes as their respective ϵ-plots given in the Fig. 4. Their relationship can be explained by (9). However, unlike ϵ, $\boldsymbol{\Psi}^T$ is independent from $\delta\psi_m$ and δr_s.

Remark 2: $\boldsymbol{\Psi}^T$ w.r.t. $\hat{\psi}_m$ is excited by n. See (15). In looking at the low derivative due to the inertia, the n-excitation can be assumed quite slow, thus, in computation of $\mathbf{L}$ for $\hat{\psi}_m$ identification, the use of steady-state form of $\boldsymbol{\Psi}^T$ given in (15) will be adequate [33].

Remark 3: $\boldsymbol{\Psi}^T$ w.r.t. $\hat{r}_s$ is excited by both n and $\hat{i}_s^r$. See (16). Dynamic counterparts of $\boldsymbol{\Psi}^T$ in (12) can offer some sort of a filtering effect in the computed $\mathbf{L}$ owing to the $\hat{T}_d, \hat{T}_q$, while yielding faster adaptation. However, to avoid oscillations in the gain, the steady-state forms can be used instead [33].

Now that the $\boldsymbol{\Psi}^T$-functions are developed, what remains is the choice of $\boldsymbol{\Psi}^T$-based algorithm. Three algorithms become relevant in this context, namely 1) stochastic gradient 2) Gauss-Newton 3) physically interpretative method, which will be discussed next. The choice of the algorithm is determined by the rate of convergence and the asymptotic accuracy they offer and at which cost of computational tediousness. Note that Λ is omitted from the respective expressions due to the same basis associated with (7).

1) Stochastic Gradient Algorithm: This is a rather simple $\mathbf{L}$-computation method given in (17) from [13]. The algorithm adopts a first-order approximation, i.e. $\boldsymbol{\Psi}^T$ to identify the search-direction with gains $\gamma[k]$ (later introduced) and $r[k]$, the scalar variant of Hessian Function. Thus, in effect, Stochastic



(a) (b)

Fig. 5: Prediction Gradient in steady-state in 4-quadrant speed-torque plane w.r.t. (a) $\hat{\psi}_m$ (b) $\hat{r}_s$.

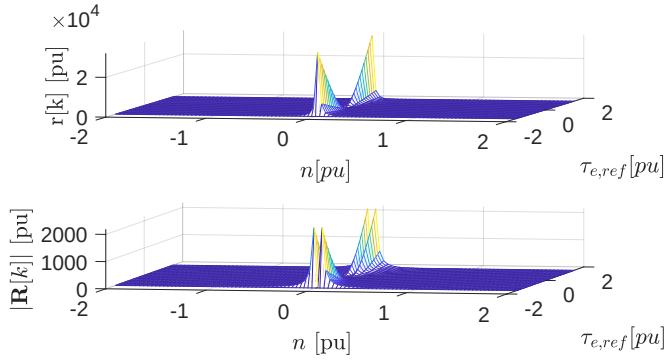


Fig. 6: Scalar Hessian, r and the determinant of Matrix Hessian, $|\mathbf{R}|$ in 4-quadrant speed-torque plane.

Gradient can be viewed as a modification to the classical gradient descent method. Close inspection of (17b) indicates that $r[k]$ is a first-order filtered version of the traces, (tr). Such filtering becomes useful, particularly when dynamic variants of $\mathbf{\Psi}^T$ are applied in computing the tr, to prevent undesirable fluctuations in the estimates.

$$\underline{\hat{\theta}}[k] = \underline{\hat{\theta}}[k-1] + \mathbf{L}[k] \cdot \underline{\epsilon}_s^r[k], \quad \mathbf{L}[k] = \gamma[k] \frac{1}{r[k]} \mathbf{\Psi}[k] \tag{17a}$$

$$r[k] = r[k-1] + \gamma[k] \Big(tr\left\{ \mathbf{\Psi}[k] \cdot \mathbf{\Psi}^T[k] \right\} - r[k-1] \Big) \tag{17b}$$

$r[k]$, in steady-state, appears as in (18).

$$tr\left\{ \mathbf{\Psi}[k] \mathbf{\Psi}^T[k] \right\} = \Big(\frac{\mathrm{d}\hat{i}_d}{\mathrm{d}\hat{\psi}_m} \Big)^2 + \Big(\frac{\mathrm{d}\hat{i}_q}{\mathrm{d}\hat{\psi}_m} \Big)^2 + \Big(\frac{\mathrm{d}\hat{i}_d}{\mathrm{d}\hat{r}_s} \Big)^2 + \Big(\frac{\mathrm{d}\hat{i}_q}{\mathrm{d}\hat{r}_s} \Big)^2 \tag{18}$$

2) Gauss-Newton Algorithm: This is, unlike the previous method, a second-order iterative minimization technique, which minimizes the criterion function more efficiently, particularly in the vicinity of the minimum. This algorithm from [13] after simplifying for the same reason associated with V_N becomes as (19).

$$\underline{\hat{\theta}}[k] = \underline{\hat{\theta}}[k-1] + \mathbf{L}[k] \cdot \underline{\epsilon}_s^r[k], \; \mathbf{L}[k] = \gamma[k] \mathbf{R}^{-1}[k] \mathbf{\Psi}[k] \tag{19a}$$

$$\mathbf{R}[k] = \mathbf{R}[k-1] + \gamma[k] \Big(\mathbf{\Psi}[k] \cdot \mathbf{\Psi}^T[k] - \mathbf{R}[k-1] \Big) \tag{19b}$$

Here, the vector form of Hessian, $\mathbf{R}[k]$ is employed. In steady-state, $\mathbf{R}[k] = \mathbf{\Psi}[k] \cdot \mathbf{\Psi}^T[k]$, where the elements of $\mathbf{R}[k]$ become as in (20).

$$\mathbf{R} = \begin{bmatrix} \Psi_{11}^2 + \Psi_{12}^2 & \Psi_{11} \cdot \Psi_{21} + \Psi_{12} \cdot \Psi_{22} \\ \Psi_{11} \cdot \Psi_{21} + \Psi_{12} \cdot \Psi_{22} & \Psi_{21}^2 + \Psi_{22}^2 \end{bmatrix} \tag{20}$$

Owning to the relatively small order of the Hessian, computation of its inverse matrix can be made convenient as in (21), by adopting an algebraic manipulation.

$$R^{-1} = \frac{1}{|\mathbf{R}|} \begin{bmatrix} R_{22} & -R_{12} \\ -R_{21} & R_{11} \end{bmatrix}$$

$$|\mathbf{R}| = \Psi_{11}^2 \Psi_{22}^2 + \Psi_{12}^2 \Psi_{21}^2 - 2 \cdot \Psi_{11} \Psi_{12} \Psi_{21} \Psi_{22} \tag{21}$$

In general, Hessian is a function of prediction gradients, and it is independent from $\underline{\epsilon}_s^r$. At zero-speed, $|\mathbf{R}|$ becomes zero so are the elements of $\mathbf{R}$ except R_{22}, thus the inverse yields zero-divided-by-zero scenarios in three of its elements. To tackle the challenge with non-existent inverse matrix due to these singularities at zero-speed, a mathematical method called Moore-Penrose pseudoinverse (MPP) is applied to find a pseudoinverse matrix which has most of the properties of $\mathbf{R}^{-1}$ [28]. To compare with SGA, the scalar- and the determinant of the matrix- Hessians, which are the denominators of the SGA and GNA, are plotted in the speed-torque plane in the Fig. 6. In connection to the GNA formulae and Fig. 6, following remarks can be made.

Remark 1: Both denominators hold similar shapes except at and around zero speed and torque. Despite the similarity in shape, $|\mathbf{R}|$ is several times smaller up to ten times at most at lower speeds, to facilitate faster adaptation with GNA in the lower speed and torque region.

Remark 2: At and around zero speed, $r = \psi_{21}^2 + \psi_{22}^2$, which is its peak. Conversely, $|\mathbf{R}|$ holds very low values (theoretically zero, but in practice, limited to very low values) to create a cleavage between the peak wedges. In summary, $|\mathbf{R}|$ is expected to offer a larger boost in gain-computation in the lower speed/torque region.

$$\mathbf{L}[k] = \frac{\gamma[k]}{|\mathbf{R}|} \begin{bmatrix} \psi_{11} R_{22} - \psi_{21} R_{12} & \psi_{12} R_{22} - \psi_{22} R_{12} \\ \psi_{21} R_{11} - \psi_{11} R_{12} & \psi_{22} R_{11} - \psi_{12} R_{12} \end{bmatrix} \tag{22}$$

From inspection of (22), all the elements in the $\mathbf{L}$ become zero at standstill. This does not influence the $\hat{\psi}_m$- adaptation as the $\underline{\epsilon}_s^r$ anyway does not carry respective information. However, $\underline{\epsilon}_s^r$ does carry information about δr_s at zero speed (if $\underline{i}_s^r \neq 0$) thus forcing L_{21}, L_{22} to null at this point, prevents possible $\hat{r}_s$-adaptation at standstill. This phenomenon indicates an inherent drawback in GNA in comparison to SGA.

3) Physically interpretative method: In this method, the estimation-gains are attempted to obtain by physically interpreting the steady-state behaviour of $\underline{\epsilon}_s^r$ in (8), s.t. $\mathbf{L} \cdot \epsilon \approx \delta\theta$. We capitalize the physical interpretations in *Remark 2* and *Remark 5* in Section III-C to identify the estimation-gains. Accordingly, $\hat{\psi}_m$ estimation-gain becomes:

$$L_{11}[k] = -\gamma[k] \cdot \hat{x}_d, \; \hat{\psi}_m[k] = \hat{\psi}_m[k-1] + L_{11}[k] \cdot \epsilon_d[k] \tag{23}$$

Similarly, the estimation gains for $\hat{r}_s$-estimation becomes as follows;

$$L_{21} = \gamma[k] \left(\frac{\hat{r}_s^2 + n^2 \cdot \hat{x}_d \cdot \hat{x}_q}{-\hat{r}_s \cdot \hat{\underline{i}}_d - n \cdot \hat{x}_q \cdot \hat{\underline{i}}_q} \right)$$

$$L_{22} = \gamma[k] \left(\frac{\hat{r}_s^2 + n^2 \cdot \hat{x}_d \cdot \hat{x}_q}{-\hat{r}_s \cdot \hat{\underline{i}}_q + n \cdot \hat{x}_d \cdot \hat{\underline{i}}_d} \right), \qquad \hat{\underline{i}}_s[k] \neq 0$$

$$\hat{r}_s[k] = \hat{r}_s[k-1] + L_{21}[k] \cdot \epsilon_d[k] + L_{22}[k] \cdot \epsilon_q[k] \qquad (24)$$

When digital implementation is concerned, SGA, GNA and PhyInt require a minimum value for their denominators (r, $|\mathbf{R}|$ or $\hat{\underline{i}}_s$) at the very low torque/speed region, in order to avoid large $\mathbf{L}$, thus to prevent noise amplification.

E. Choice of Gain Sequence and Initial Values

Gain-sequence, γ can be viewed as a memory-coefficient. Larger γ enables faster tracking by 'forgetting' the older ϵ in preference to the more recent ones however, at the expense of increased noise sensitivity. In the context of tracking slowly varying parameters, it is shown in [13] that $\gamma[k]$ is often chosen to be a constant, γ_0, which can be expressed as follows;

$$\hat{\underline{\theta}}[k] = \hat{\underline{\theta}}[k-1] + \frac{T_{samp}}{T_0} \cdot \mathbf{\Psi}[k] \cdot \underline{\epsilon}[k], \quad \gamma_0 = \frac{T_{samp}}{T_0} \qquad (25)$$

Thus, γ_0 is nothing but an integral time constant, in which, T_0 is, in fact, the chosen variable. T_0 should be chosen such that the estimated parameters are almost constant over a period of length T_0. When temperature-sensitive parameters are concerned, T_0 could be in the range of a few seconds such that it still produces a fast enough algorithm to track slow-varying parameters yet not too fast to prevent being sensitive to noise.

Having as much accurate initial values can circumvent a fundamental challenge with the gradient-based minimization algorithms that can be mislead by local minima. By using offline methods and identification runs during the commissioning, initial machine parameters can be identified accurately.

IV. Gain-Scheduling Scheme

In this section, the impact of the simultaneous adaptation that showcases the requirement of having a decoupling mechanism will be analyzed. Subsequently, the implementation of the gain-scheduling scheme is disclosed.

A. Impact of simultaneous adaptation

The possibility of simultaneous adaptation can be favourable in many ways. However, it was emerged in the *Remark 5* in the Section III-C, the sensitivity of $\underline{\epsilon}$, hence $\mathbf{\Psi}^T$ w.r.t. $\hat{\psi}_m$ and $\hat{r}_s$ is prominent in exclusive speed regions. This suggests that the $\hat{\psi}_m$-adaptation will be successful mostly beyond very low speeds, whereas the $\hat{r}_s$-adaptation during very low speeds. These sensitivities are illustrated in the Fig. 7, in which we see there are also overlapping regions. In this view, it is interesting to examine how an error in one parameter influences the other, when the rotor speed is in a speed zone where the erroneous parameter is less sensitive, but the other is.

The influence from δr_s on the $\hat{\psi}_m$ is first investigated. Due to the influence from the q-component on $\hat{\psi}_m$-adaptation is negligible, the analysis is performed only in relation to the d-component. Accordingly, from (8), it can be derived an expression as in (26) for the unfairly adapted $\hat{\psi}_m$ in steady state due to δr_s.

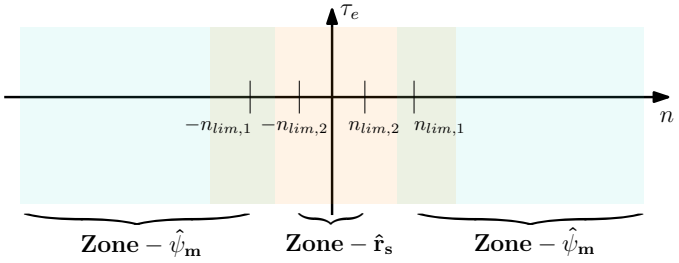


Fig. 7: Illustration of zonal adaptation of $\hat{r}_s$ and $\hat{\psi}_m$ implemented using the gain-scheduling scheme.

$$\hat{\psi}_m = \psi_m + \frac{(\hat{r}_s \cdot i_d + n \cdot \hat{x}_q \cdot i_q)}{n^2 \cdot \hat{x}_q} \cdot \delta r_s \approx \psi_m + \frac{i_q}{n} \cdot \delta r_s \qquad (26)$$

It's worth noting that in a typical IPMSM, $\psi_m \gg r_s$, $r_s \ll 1$ in pu. Under this context, what the expression says is that the impact of δr_s becomes significant only when $n \ll 1$. To eliminate this undue impact during very low speeds, a requirement to halt the $\hat{\psi}_m$-adaptation arises, at very low speeds.

Similarly, the influence from $\delta\psi_m$ on the $\hat{r}_s$ can be analyzed. Unlike the previous discussion, here we should consider the influence from both ϵ_d, ϵ_q, thus two expressions as in (27) can be derived from (8).

$$\hat{r}_s = \; r_s + \frac{n^2 \cdot \hat{x}_q}{(\hat{r}_s \cdot i_d + n \cdot \hat{x}_q \cdot i_q)} \cdot \delta\psi_m$$

$$\hat{r}_s = \; r_s + \frac{n \cdot \hat{r}_s}{(n \cdot \hat{x}_q \cdot i_d - \hat{r}_s \cdot i_q)} \cdot \delta\psi_m \qquad (27)$$

In view of $\psi_m \gg r_s$, $r_s \ll 1$, what these expressions reveal is that as the rotor speed increases, the impact from $\delta\psi_m$ dramatically increases to unduly adapt $\hat{r}_s$, thus it signifies a mandatory requirement to cut-off the $\hat{r}_s$-adaptation beyond very low speeds.

B. Gain-Scheduler

Based on the emerged requirements, a gain-scheduler is proposed to limit the respective adaptations to their own sensitive speed zones as shown in the Fig. 7. Such zonal adaptation is not only encouraged by the physical behaviour of the sensitivies but also becomes mandatory to eliminate undue adaptations. Thus the respective gains are scheduled as given in (28), where $x = 1, 2$ and $|n_{lim,1}| > |n_{lim,2}|$.

$$L_{1,x} = \begin{cases} L_{1,x}, |n| > |n_{lim,1}| \\ 0, \; otherwise \end{cases}, L_{2,x} = \begin{cases} L_{2,x}, |n| < |n_{lim,2}| \\ 0, \; otherwise \end{cases} \qquad (28)$$

V. Real-Time Simulation Based Validation

In this section, we attempt to make a choice among the three $\mathbf{\Psi}^T$-based algorithms with the aid of a Xilinx Zynq System on Chip-based ERTS. $\hat{\psi}_m$ and $\hat{r}_s$ are identified online when the respective physical values undergo a step-change of -8%. The model inductances are updated online using the offline identified values, thus assumed no discrepancy between the model

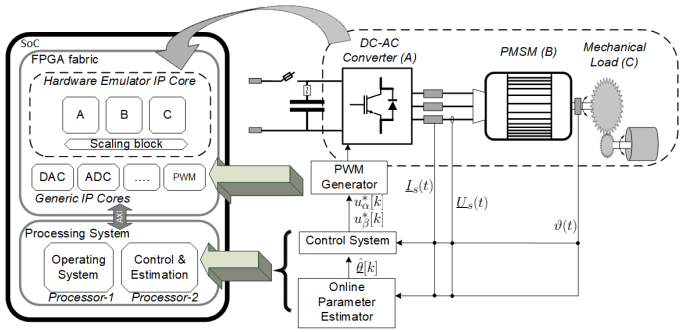


Fig. 8: Overview of the Embedded Real-Time Simulator designed for simulation and implementation of three-phase IPMSM Drive.

TABLE I: Parameters of the Experimental Plant.

Symbol	Parameter	Value
U_n	IPMSM Rated Voltage	400 V
I_n	IPMSM Rated Current	4.93 A
P_n	IPMSM Rated Power	3 kW
N_n	IPMSM Rated Speed	1000 rpm
T_n	IPMSM Rated Torque	32.6 Nm
p	IPMSM Number of pole-pairs	3
R_s	Stator Resistance (offline)	2.25 Ω
Ψ_m	Permanent magnet flux linkage (offline)	1.14 Wb
L_d	IPMSM d-axis inductance (rated load)	0.0953 H
L_q	IPMSM q-axis inductance (rated load)	0.206 H
U_{dc}	DC bus voltage	220 V
f_{sw}	Power device switching frequency	4 kHz
T_{samp}	Sampling period	125 μs

and physical counterparts. A step-change in motor parameters allows us to assess the stability and the tracking speed of the proposed method, despite it is unusual for temperature-sensitive parameters. The overview of the ERTS is illustrated in the Fig. 8. The power hardware components of the drive are programmed in the Field-Programmable Gate Array (FPGA) fabric of the SoC to achieve real-time emulation at a time-step of 1 μs. The control, state- and parameter- estimation algorithms and likewise relatively slower processes are programmed in the on-chip processor at the PWM double-update time-step of 125 μs. The validation of this ERTS against the Matlab/Simulink based offline simulation is given in [34]. Two-level VSI with asymmetrical modulation and 3rd harmonic injection is used to drive the machine. A speed-dependent gain-scheduler is applied to restrain the $\hat{r}_s$-adaptation between -10 to 10 rpm and $\hat{\psi}_m$-adaptation beyond $|100|$ rpm. Table I tabulates the experimental plant data. To avoid the oscillations in the adaptation gains, the steady-state forms of the $\mathbf{\Psi}^T$ is used where applicable [33]. The respective gain-sequence values for $\hat{\psi}_m$ and $\hat{r}_s$ -adaptation using SGA, GNA and PhyInt are tabulated in the Table II. These values are chosen in order to demonstrate comparable, yet sufficiently rapid tracking performances between the two different algorithms.

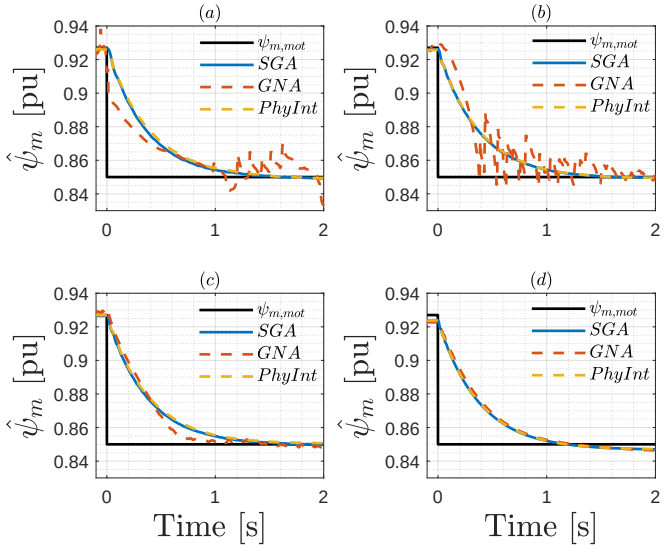


Fig. 9: $\hat{\psi}_m$ online adaption with SGA, GNA and PhyInt (a) $n = -0.2\,pu$, $\tau_{el} = 0\,pu$ (b) $n = -0.4\,pu$, $\tau_{el} = 0.2\,pu$ (c) $n = 0.4\,pu$, $\tau_{el} = 0.2\,pu$ (d) $n = 0.8\,pu$, $\tau_{el} = 0.4\,pu$.

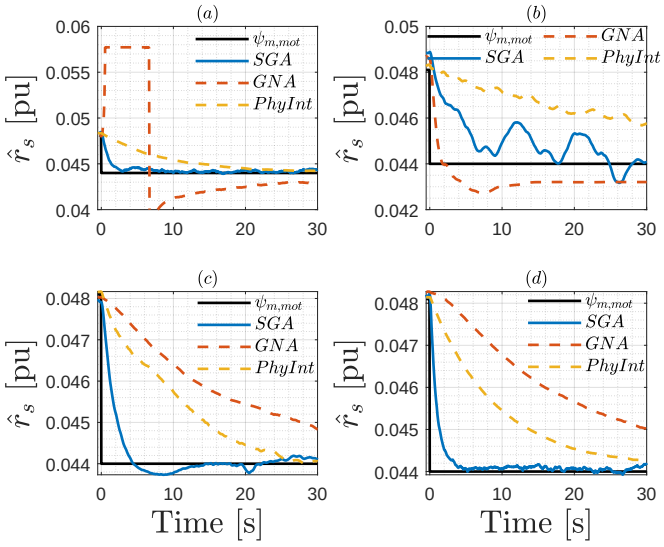


Fig. 10: $\hat{r}_s$ online adaption with SGA, GNA and PhyInt (a) $n = -0.05\,pu$, $\tau_{el} = 0.2\,pu$ (b) $n = 0\,pu$, $\tau_{el} = 0.2\,pu$ (c) $n = 0\,pu$, $\tau_{el} = 0.6\,pu$ (d) $n = 0.05\,pu$, $\tau_{el} = 0.6\,pu$.

TABLE II: Gain-sequences for online estimation.

Symbol	Parameter	γ_0[pu] SGA, PhyInt	γ_0[pu] GNA
	For $\hat{\psi}_m$-estimation		
$\gamma_{0,rk}$	Gain-sequence for Hessian	6.25×10^{-4}	6.25×10^{-4}
$\gamma_{0,Lk}$	Gain-sequence for Gain	3.25×10^{-4}	3.25×10^{-4}
	For $\hat{r}_s$-estimation		
$\gamma_{0,rk}$	Gain-sequence for Hessian	6.25×10^{-4}	6.25×10^{-5}
$\gamma_{0,Lk}$	Gain-sequence for Gain	6.25×10^{-5}	7.5×10^{-6}

Fig. 9 contains ψ_m online tracking trajectories overlaid when the three algorithms are adopted at different speeds and loads. Fig. 9 (a) and (b) are when the rotor speeds are negative. In case (a) the load-torque τ_{el} is zero and in (b), τ_{el} = 0.2 meaning, the machine will be in generating mode, to see

TABLE III: Processor execution times of the different algorithms.

Algorithm	Execution Time (μs)
SGA	∼5.2
GNA	∼5.6
PhyInt	∼5.0

nearly no torque in the shaft. Under these conditions, both SGA and PhyInt yield stable and noise-free tracking. GNA, too, succeeds in convergence, yet seem to be overly excited along the way. At low loads, the $\mathbf{R}-$elements in (21) become very small which can excessively boost $\mathbf{L}$. This effect is what causes the oscillations in the GNA-trajectories in (a) and (b). At higher loads as in the Fig. 9 (c) and (d), GNA yields smoother adaptation like the SGA and PhyInt.

Similarly, the R_s-adaptation related to the three algorithms is presented in the Fig. 10. In this case, to achieve stable tracking with GNA, γ_0 needed to be made nearly 10 times smaller than that of SGA or PhyInt. This hinders the GNA- tracking speed as it is made evident in all cases. PhyInt, on the other hand, while offering noise-free tracking, the convergence speed is significantly lower in comparison to SGA.

In general, SGA and PhyInt display more stable adaptation consistently. They become the same in steady-state, if $r[k]$ in the SGA is computed using only the respective prediction-gradient instead of the full trace as given in (18). One advantage with SGA over PhyInt is the use of dynamic $r[k]$ (17b) allows initialization and the choice of γ_{rk}, that can determine the magnitude and length of adaptation-boosting. This facilitates faster and filtered estimations, particularly at start of the routines. In comparison to the execution times of the three algorithms, there is very little to differentiate between the algorithms, as one can see in the Table III, owing to the powerful floating-point processor in the SoC. The computational burden posed by GNA can however be increased when the rank of $\mathbf{L}$ increases, which will involve the demanding task of real-time matrix inversion.

VI. Experimental Validation

Here, the $\mathbf{\Psi}^T$-based RPEM algorithms for parameter identification are attempted to validate using an experimental setup shown in Fig. 11 of which the data is given in the Table I. It was evident in the previous section that PhyInt can be viewed as a less flexible variant of SGA, thus, it will be omitted in this experimental validation. The same digital controller that houses the ERTS is used to control the motor drive setup. The γ_0-values tabulated in Table II are applied here.

A. Ψ_m-Tracking Validation

During the experiments, it was identified that the dynamic forms of the ψ_{21} and ψ_{22} cause to superimpose the current-sensor noise in the GNA-based tracking trajectories, particularly at the event of no-load. This could have been mitigated by using a 100 times smaller $\gamma_{0,Lk}$ for GNA than what is tabulated in II, however at the price of slower convergence. Also, these oscillations disappear as soon as the IPMSM is loaded. Instead, in order to achieve a comparable convergence speed, the steady-state forms of ψ_{11} and ψ_{12} are chosen in both SGA and GNA computations. The performance of the online adaptation of $\hat{\psi}_m$ using these algorithms at various rotor speeds and load torques are plotted in Fig. 12. The reference (Ref) in the plot is the offline identified $\psi_m = 0.895\,pu$. The no-load adaptation with GNA is slightly quicker than that with the SGA, at the price of a 6% overshoot, as per Fig. 12(a). When the IPMSM is loaded with 0.4 pu load-torque, the adaptation between the algorithms is nearly identical as seen in Fig. 12(b). Irrespective of the load, at the given speed, the convergence occurs within 2 seconds which is sufficient for a temperature-induced ψ_m-variation.

Fig. 12(c) and (d) show how the $\hat{\psi}_m$ behaves upon a step-change in the speed reference and load-torque respectively. In the first case, the speed varies from -0.3 to +0.3 pu speed, during which $\hat{\psi}_m$ remains stable. When a step-change in the load-torque occurs from -0.4 pu to +0.4 pu, i.e. when the sign of the i_q changes, again the $\hat{\psi}_m$ remains stable with the SGA. When GNA is concerned, the $\hat{\psi}_m$ oscillates when the rotor speed is unsettled, yet converges afterward. A summary of the performance is tabulated in the Table IV.

B. R_s-Tracking Validation

As in the previous case, the steady-state forms of the ψ_{21} and ψ_{22} are incorporated when GNA gains are computed. The respective experimental validations are in Fig. 13. The adaption performances at standstill and at 0.005 pu speed are in Fig. 13 (a) and (b) respectively when the load-torque is 0.4 pu. In both cases, the performance differences between the algorithms are marginal. The convergence performances upon a speed reference and load-torque step-change are plotted in the 13(c) and (d) respectively. Despite the steady-state behaviors being indistinguishable, it is seen that the SGA yields more stable tracking during the load (thus the rotor-speed) transient. At low speeds, a speed ripple is evident in the rotor shaft which is superimposed on the estimate-trajectories as seen in the 13(b) and (c).

A summary of the performance is tabulated in the Table IV. The time taken by the SoC to process SGA and GNA routines is nearly the same as given in the Table III which is well within the interrupt service routine for IGBT-based drives, thus the computational burden is not a matter of concern.

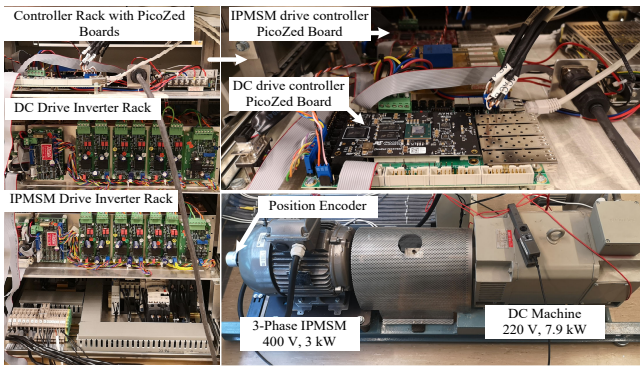


Fig. 11: Experimental Setup having IPMSM as the drive machine and DC machine as the load machine mounted on the same shaft.

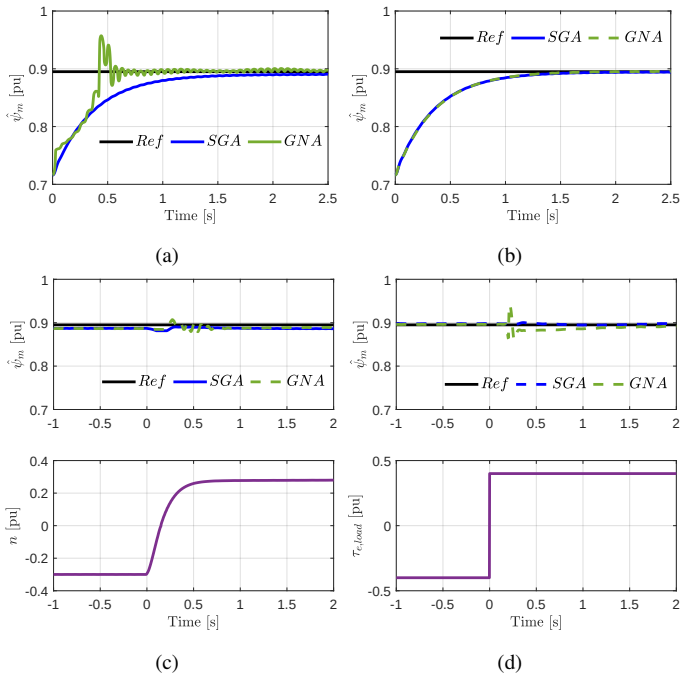


Fig. 12: Experimental validations of $\hat{\psi}_m$ -online adaptation with SGA and GNA when (a) no-load at 0.3 pu speed (b) 0.4 pu load-torque at 0.3 pu speed (c) speed reference step-change from -0.3 to 0.3 at 0.4 pu load-torque (d) load step-change from -0.4 to +0.4 pu load-torque at 0.3 pu speed.

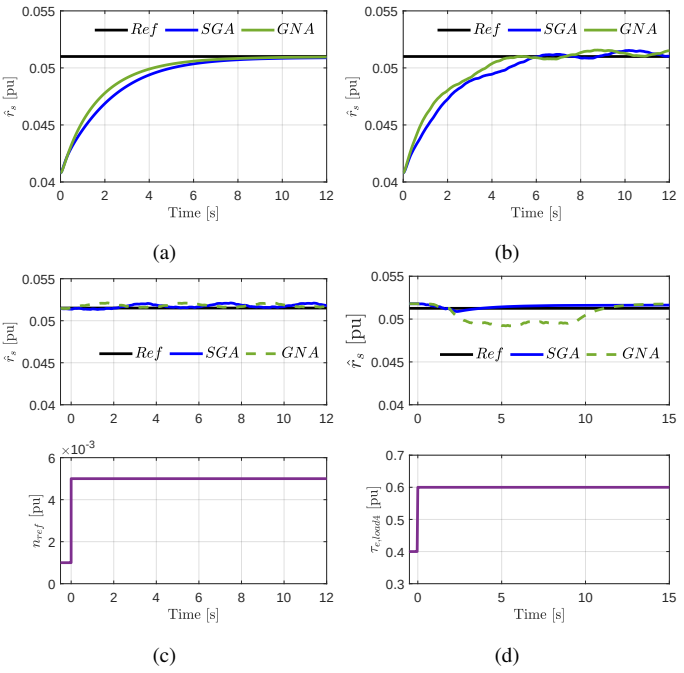


Fig. 13: Experimental validations of $\hat{r}_s$ -online adaptation with SGA and GNA when (a) 0.4 pu load-torque at standstill (b) 0.4 pu load-torque at 0.005 pu speed (c) speed reference step-change from 0.001 to 0.005 pu at 0.4 pu load-torque (d) load step-change from 0.4 to 0.6 pu load-torque at standstill.

TABLE IV: Performance comparison summary between SGA and GNA.

Case	SGA	GNA
W.r.t. $\hat{\psi}_m$-estimation		
Convergence speed ($\tau_l = 0, n = 0.3\,pu$)	$\sim$2 s	$\sim$0.5 s
Convergence speed ($\tau_l = 0.4, n = 0.3\,pu$)	$\sim$1.5 s	$\sim$1.5 s
Steady-state error ($\tau_l = 0, n = 0.3\,pu$)	(-)0.5%	0.5%
Steady-state error ($\tau_l = 0.4, n = 0.3\,pu$)	$\sim$0%	$\sim$0%
W.r.t. $\hat{r}_s$-estimation		
Convergence time ($\tau_l = 0.4, n = 0\,pu$)	8 s	8 s
Convergence time ($\tau_l = 0.4, n = 0.005\,pu$)	6 s	4 s
Steady-state error ($\tau_l = 0.4, n = 0\,pu$)	0	0
Steady-state error ($\tau_l = 0.4, n = 0.005\,pu$)	$\sim$0	$\sim$0

VII. Conclusion

This article proposed a prediction-gradients-assisted RPEM-based framework and three algorithms to identify parameters of electric machines, and the methods are demonstrated and validated using an IPMSM by identifying temperature-sensitive parameters online. The predictor is arranged in an open-loop thus the prediction error is enriched with parametric errors, a feature that is exploited by deriving prediction gradients, that becomes the main element in the estimation gains in this context. With the aid of real-time simulation, a performance comparison of the three algorithms is executed across the operating range. Experimental results show that both the SGA and GNA offer reasonable tracking performance. Despite the latter can offer faster tracking in principle, it becomes overly excited at lower torque/speed region and inherently prevents R_s-tracking at zero-speed, unlike the other two methods. Moreover, very fast adaptation has little use when the large thermal time constants are concerned. Due to the attributes of the dynamic hessian, SGA offers controllable tracking speeds at the start, unlike the PhyInt. Given the stable, flexible, consistent performance and the simplicity in the implementation, RPEM with SGA can be a practical solution for temperature-sensitive parameter estimation of electrical machines.

Acknowledgement

Authors are thankful to the support received from the service lab/workshop of the Department of Electric Power Engineering of NTNU in developing the ERTS and building the experimental setup. Associate Professor Pål Olsen is acknowledged for providing the WEG IPMSM for tests. The support from the Associate Professor Jonas Nøland in formulating this paper is also acknowledged with gratitude.

References

[1] E. Bostanci, M. Moallem, A. Parsapour, and B. Fahimi, "Opportunities and challenges of switched reluctance motor drives for electric propulsion: a comparative study," *IEEE Transactions on Transportation Electrification*, vol. 3, no. 1, pp. 58–75, 2017.

[2] Z. Q. Zhu, D. Liang, and K. Liu, "Online parameter estimation for permanent magnet synchronous machines : an overview," *IEEE Access*, vol. 9, pp. 59059–59084, 2021.

[3] M. S. Rafaq and J. W. Jung, "A Comprehensive review of state-of-the-art parameter estimation techniques for permanent magnet synchronous motors in side speed range," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 7, pp. 4747–4758, 2020.

[4] P. Vaclavek, P. Blaha, and I. Herman, "AC drive observability analysis," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 8, pp. 3047–3059, 2013.

[5] S. J. Underwood and I. Husain, "Online parameter estimation and adaptive control of permanent-magnet synchronous machines," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 7, pp. 2435–2443, 2010.

[6] Q. Wang, G. Wang, N. Zhao, G. Zhang, Q. Cui, and D. Xu, "An impedance model-based multiparameter identification method of PMSM for both offline and online conditions," *IEEE Transactions on Power Electronics*, vol. 36, no. 1, pp. 727–738, 2021.
[7] W. Xu and R. D. Lorenz, "High-frequency injection-based stator flux linkage and torque estimation for DB-DTFC implementation on IPMSMs considering cross-saturation effects," *IEEE Transactions on Industriy Applications*, vol. 50, no. 6, pp. 3805–3815, 2014.
[8] U. Schaible and B. Szabados, "Dynamic motor parameter identification for high speed flux weakening operation of brushless permanent magnet synchronous machines," *IEEE Transactions on Energy Conversion*, vol. 14, no. 3, pp. 486–492, 1999.
[9] S. W. Su, C. M. Hackl, and R. Kennel, "Analytical prototype functions for flux linkage approximation in synchronous machines," *IEEE Open Journal of the Industrial Electronics Society*, vol. 3, no. March, pp. 265–282, 2022.
[10] S. Stipetic, M. Kovacic, Z. Hanic, and M. Vrazic, "Measurement of excitation winding temperature on synchronous generator in rotation using infrared thermography," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 5, pp. 2288–2298, 2012.
[11] D. Fernandez, D. Reigosa, T. Tanimoto, T. Kato, and F. Briz, "Wireless permanent magnet temperature & field distribution measurement system for IPMSMs," *2015 IEEE Energy Conversion Congress and Exposition, ECCE 2015*, pp. 3996–4003, 2015.
[12] O. Wallscheid, "Thermal monitoring of electric motors: state-of-the-art review and future challenges," *IEEE Open Journal of Industry Applications*, vol. 2, no. June, pp. 204–223, 2021.
[13] L. Ljung and T. Soderstrom, *Theory and Practice of Recursive Identification*. Cambridge, Massachusetts: The MIT Press, 2nd ed., 1985.
[14] A. Brosch, S. Hanke, O. Wallscheid, J. Böcker, and S. Member, "Data-driven recursive least squares estimation for model predictive current control of permanent magnet synchronous motors," *IEEE Transactions on Power Electronics*, vol. 36, no. 2, pp. 2179–2190, 2021.
[15] S. Morimoto, M. Sanada, and Y. Takeda, "Mechanical sensorless drives of IPMSM with online parameter identification," *IEEE Transactions on Industry Applications*, vol. 42, no. 5, pp. 1241–1248, 2006.
[16] Y. Inoue, Y. Kawaguchi, S. Morimoto, and M. Sanada, "Performance improvement of sensorless IPMSM drives in a low-speed region using online parameter identification," *IEEE Transactions on Industry Applications*, vol. 47, no. 2, pp. 798–804, 2011.
[17] L. Harnefors and H. P. Nee, "Adaptive sensorless control of induction motors for improved low-speed performance," *Conference Record - IAS Annual Meeting (IEEE Industry Applications Society)*, vol. 1, pp. 278–285, 1996.
[18] X. Li and R. Kennel, "General formulation of kalman-filter-based online parameter identification methods for VSI-fed PMSM," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 4, pp. 2856–2864, 2021.
[19] T. Soderstrom and P. Stoica, *System Identification*. Uppsala: Prentice Hall International, 2001.
[20] H. Børsting, M. Knudsen, H. Rasmussen, and P. Vadstrup, "Estimation of mechanical signals in induction motors using the recursive prediction error method," in *EPE*, pp. 455–460, 1993.
[21] D. Ljungquist and J. G. Balchen, "Recursive prediction error methods for online estimation in nonlinear state-space models," *Proceedings of the IEEE Conference on Decision and Control*, vol. 1, pp. 714–719, 1993.
[22] A. Piippo, M. Hinkkanen, and J. Luomi, "Adaptation of motor parameters in sensorless PMSM drives," *IEEE Transactions on Industry Applications*, vol. 45, no. 1, pp. 203–212, 2009.
[23] S. S. Badini and V. Verma, "A new stator resistance estimation technique for vector-controlled PMSM drive," *IEEE Transactions on Industry Applications*, vol. 56, no. 6, pp. 6536–6545, 2020.
[24] Z. H. Liu, H. L. Wei, Q. C. Zhong, K. Liu, X. S. Xiao, and L. H. Wu, "Parameter estimation for VSI-Fed PMSM based on a dynamic PSO with learning strategies," *IEEE Transactions on Power Electronics*, vol. 32, no. 4, pp. 3154–3165, 2017.
[25] Z. Wang, J. Chai, X. Xiang, X. Sun, and H. Lu, "A novel online parameter identification algorithm designed for deadbeat current control of the permanent-magnet synchronous motor," *IEEE Transactions on Industry Applications*, vol. 58, no. 2, pp. 2029–2041, 2022.
[26] S. Zhao, F. Blaabjerg, and H. Wang, "An overview of artificial intelligence applications for power electronics," *IEEE Transactions on Power Electronics*, vol. 36, no. 4, pp. 4633–4658, 2021.
[27] R. Nilsen and M. P. Kazmierkowski, "Reduced-order observer with parameter adaption for fast rotor flux estimation in induction machines," *IEE Proceedings*, vol. 136, no. 1, pp. 35–43, 1989.
[28] A. Perera and R. Nilsen, "Gauss-Newton: A prediction-error-gradient based algorithm to track PMSM parameters online," in *9th IEEE International Conference on Power Electronics, Drives and Energy Systems, PEDES 2020* (IEEE, ed.), (Jaipur, India), 2020.
[29] H. Buhler, *Introduction to the Theory of Controlled Three-Phase Drives. Volume 1: Basics. Volume 2: Applications*. Basel: Birkhauser, 6 ed., 1977.
[30] P. Weichbold and T. v. Raumer, "Influence of dead time effects of PWM-VSI on current control," in *EPE* (IEEE, ed.), (Graz), p. 8, 2001.
[31] S. Morimoto, M. Sanada, and Y. Takeda, "Wide-speed operation of interior permanent magnet synchronous motors with high-performance current regulator," *IEEE Transactions on Industry Applications*, vol. 30, no. 4, pp. 920–926, 1994.
[32] J. Lambert, *Computational Methods in Ordinary Differential Equation*. Hoboken: John Wiley & Sons, Inc, 1973.
[33] A. Perera and R. Nilsen, "A recursive prediction error method with effective use of gradient-functions to adapt PMSM parameters online," in *IEEE Industry Applications Society Annual Meeting*, (Detroit, USA), pp. 2–6, IEEE, 2020.
[34] A. Perera, R. Nilsen, T. Haugan, and K. Ljøkelsøy, "A design method of an embedded real-time simulator for electric drives using low-cost system-on-chip platform," in *PCIM Europe digital days*, (Nuremberg, Germany), 2021.

Aravinda Perera was born in Colombo, Sri Lanka, in 1984. He received the B.Sc. (Honours) degree from University of Moratuwa, Sri Lanka and M.Sc. degree from Norwegian University of Science and Technology (NTNU), Trondheim in 2009 and 2012 respectively, both in electrical engineering. Aravinda has several years of industrial experience in marine, and offshore oil & gas industries. From 2012 to 2018, Aravinda was employed at Siemens, Norway in R&D and Engineering divisions as a Senior Engineer and Technical Project Manager, during which he authored four patents. From 2018, he reads a Ph.D. degree in NTNU focusing sensorless control methods for electric drive trains in deep-sea mining vehicles. His research interests include electric drives, wide band-gap devices and embedded control methods.

Roy Nilsen received his M.Sc. in 1984 and PhD in 1987 from Norwegian Institute of Technology (NTH). In the period 1987–1996 he was employed at ABB. In 1988/89 he worked at ABB Drives in Turgi, Switzerland. From 1989 until 1996 he was with ABB Corporate Research in Oslo, Norway. In the period 1996–2006 he was professor in Electric Drives at the Norwegian University of Science and Technology (NTNU). From 2001 until 2017 he has been participating in developing drives for marine applications in Aker/Wartsila/Thc Switch. Since April 2017 he is back as professor in Electric Drives at the Department of Electric Power Engineering, NTNU. He also serves as the head of the Power Electronic Systems and Components (PESC) group of the department.