A Framework and an Open-Loop Method to Identify PMSM Parameters Online

Aravinda Perera, Roy Nilsen
Department of Electric Power Engineering, Norwegian University of Science and Technology, Trondheim, Norway

***Abstract*-- A method for online adaptation of electric parameters of a rotating machine is proposed herein. The concept adopts the recursive prediction error method (RPEM) for parameter adaptation, that exploits the prediction-error gradient functions ($\mathbf{\Psi}^T$). With the aim of setting a general framework for the cause, the method is systematically demonstrated for online identification of permanent magnet flux linkage (Ψ_m) and stator-winding resistance (R_s) of an interior permanent magnet synchronous machine (IPMSM). Additionally, an experiment to estimate R_s at the start-up is presented. The gain-matrix is identified using the stochastic gradient algorithm (SGA). Simulation results validate the rapid convergence performance, adaptability and tuning flexibility of the proposed method.**

***Index Terms*— Hessian, parameter sensitivity, gain scheduling, stochastic gradient, variable speed drives**

I. Introduction

More and more mission-critical engineering applications as such as aerospace, offshore oil and gas and seabed mineral mining are embracing electric machinery over the traditional mechanical, hydraulic or pneumatic counterparts. IPMSM is a popular candidate in such high-power applications owing to its superior efficiency and torque-density.

The operating conditions are often harsh in such industrial drives where the ambient temperature can be sometimes several folds of the room temperature, which can affect the temperature-sensitive motor parameters, i.e. R_s and Ψ_m. The motor parameters, on the other hand, influence the control of the electric drive. Moreover, mechanical sensor-less control systems have been state-of-the-art in the applications that demand high robustness, in which it is common to employ field excitation (FE) -based methods to estimate the rotor position, particularly beyond zero and very-low speeds. Such FE methods are also heavily dependent on the machine parameters, therefore, unaccounted changes of the machine parameters in the control system can result in erroneous position estimation, consequently, poor torque and speed -control . Despite winding inductances, particularly in the (fictitious) quadrature axis (L_q), influence the rotor-position estimation, it is reasonable to adapt L_q with the aid of a current or flux based function or a simpler offline experiment [1]. Therefore, when high performance, mission critical applications are concerned, R_s and Ψ_m should be adapted online. Several online parameter estimation techniques have been reviewed in [2] and [3] in which MRAS [1], [4], Kalman Filter [5] and recursive least square (RLS) [6], [7] -based methods appear to be the common approaches. In looking at the parameter-error-sensitivity, convergence, implementation complexity and computational burden, each of these methods have their own pros and cons.

This paper presents an online parameter estimator (OPE) as in Fig. 1, which is highly sensitive to parametric mismatches between the estimation and physical quantities. This sensitivity is capitalized by the parameter adaptation algorithm (PAA) to recursively estimate Ψ_m and R_s of IPMSM. The PAA is premised on the RPEM explained and applied in [8] and [9] respectively. In order to search the parameters in the defined parameter-space, a sub-algorithm known as stochastic gradient algorithm (SGA) is applied. The SGA exploits the sensitivity of the prediction-errors against the varying parameters in its cause. This sensitivity is termed as prediction-error gradient (PEG) and denoted by $\mathbf{\Psi}^T$ as in [8]. The concept is developed systematically by following the step-by-step approach in [8] which enables to identify gains for the parameter adaptations analytically. Despite the method is demonstrated for an IPMSM drive, its general framework is applicable for any type of electric drive.

II. Motor & Estimation Models

A. IPMSM Mathematical Model

The mathematical model of the electrical part of the machine is in the rotor co-ordinates, when given in the per-unit (pu) system:

$$\underline{u}_s^r = r_s \cdot \underline{i}_s^r + \frac{1}{\omega_n}\frac{d\underline{\psi}_s^r}{dt} + \mathbf{j}\cdot f_k \cdot \underline{\psi}_s^r, \quad \underline{\psi}_s^r = \mathbf{x}_s^r \cdot \underline{i}_s^r + \underline{\psi}_m^r$$
$$\underline{i}_s^r = [i_d \quad i_q]^T \quad \underline{\hat{\psi}}_m^r = [\hat{\psi}_m \quad 0]^T \quad \mathbf{x}_s^r = \begin{bmatrix} x_d & 0 \\ 0 & x_q \end{bmatrix} \quad \mathbf{j} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \tag{1}$$

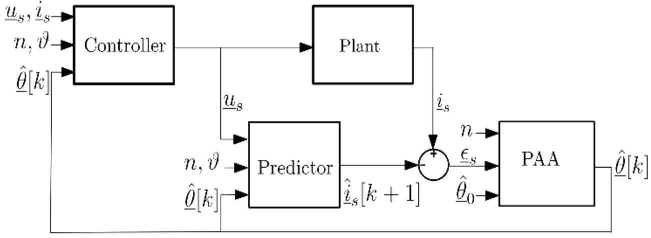

Fig. 1. Proposed online parameter estimation method block diagram

[1]This work is supported by the NTNU Oceans pilot program on deep-sea mining.

Here, ω_n is the nominal rotational frequency. ϑ is the electrical angle of the mechanical position $p*\vartheta_{mech}$, where p is the number of pole pairs. Electrical speed is denoted by n. The superscript and subscript denote the reference frame and the location of the quantity (s-stator, r-rotor, m-magnet) respectively.

B. Online Parameter Estimation Model

The online parameter estimation (OPE) model in Fig. 1 has an open-loop structure where $\underline{\varepsilon}_s$ (prediction error) is not fed back to the predictor for immediate error-correction. Thus, this model becomes highly sensitive to parameter estimation mismatches with reference to actual physical quantities. This is the very intention behind the selection of such an open-loop model in this cause, because these sensitivities will be profoundly exploited in the proposed PAA. Full-order model ($\mathscr{M}_{u\vartheta}$), is used in this paper with stator currents chosen as state variables. The rotor-oriented model is chosen for current prediction in the predictor.

$$\underline{u}_s^r = \hat{r}_s \cdot \hat{\underline{i}}_s^r + \frac{\mathbf{x}_s^r}{\omega_n} \cdot \frac{d\hat{\underline{i}}_s^r}{dt} + \mathbf{j} \cdot n \cdot \hat{\mathbf{x}}_s^r \cdot \hat{\underline{i}}_s^r + \mathbf{j} \cdot n \cdot \hat{\underline{\psi}}_m^r$$

$$\hat{\underline{\theta}} = \left[\hat{\psi}_m \quad \hat{r}_s\right]^T \qquad (2)$$

$$\underline{i}_s^r = \mathbf{T}_{ss}^r(\vartheta) \cdot \underline{i}_s^s \qquad \underline{u}_s^r = \mathbf{T}_{ss}^r(\vartheta) \cdot \underline{u}_s^s$$

Here, from the estimated parameter matrix, $\hat{\underline{\theta}}$, $\hat{x}_d, \hat{x}_q$ are omitted to curtail the discussion only to scope of interest. As shown in (2), position and speed become inputs in the model, thus, they must be either measured or estimated. In this paper, a position sensor is assumed. In [10], this OPE is extended to sensorless control of IPMSM.

$\mathscr{M}_{u\vartheta}$ is a second order system and the eigenvalues of this model are speed dependent. The system matrix $\mathbf{A}$ of the system can be expressed as:

$$\lambda \cdot I_2 - \mathbf{A} = \begin{bmatrix} \lambda + \dfrac{\omega_n \cdot \hat{r}_s}{\hat{x}_d} & -\dfrac{\omega_n \cdot n \cdot \hat{x}_q}{\hat{x}_d} \\ \dfrac{\omega_n \cdot n \cdot \hat{x}_d}{\hat{x}_q} & \lambda + \dfrac{\omega_n \cdot \hat{r}_s}{\hat{x}_q} \end{bmatrix} \qquad (3)$$

The eigenvalues become:

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

$$T_d = \frac{x_d}{r_s \cdot \omega_n}, T_q = \frac{x_q}{r_s \cdot \omega_n}$$

III. Parameter Sensitivity of the Control Strategy

Maximum-torque-per-ampere (MTPA) -control strategy is considered in this work. Accordingly, the optimal d- and q-current references are calculated by the help of the 3rd order expression given in [11]. It is interesting to firstly identify the sensitivity of the torque to incorrect model parameters under this control strategy. It turns out that the effect of misestimated r_s does not affect the torque control unless due to voltage limitation of the inverter during the field weakening range. The incorrect value of ψ_m, however, has an inevitable influence in the torque control in the complete torque-speed plane. As in Fig. 2 where a 10% under-estimated ψ_m has been considered for current reference computation in the controller.

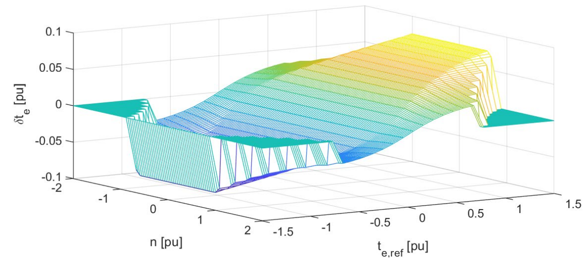

Fig. 2. Torque error due to 10% under-estimated ψ_m

IV. Criterion and Prediction Error -Functions

A quadratic criterion with a stator currents-based prediction errors are chosen to develop the proposed estimation method. The continuous version of the PAA then becomes, where $\mathbf{L}_c$ is the continuous gain matrix:

$$\underline{\varepsilon}_s^r(t,\hat{\underline{\theta}}) = \delta \underline{i}_s^r = \underline{i}_s^r - \hat{\underline{i}}_s^r \qquad \underline{i}_s^r = \mathbf{T}_{ss}^r(\vartheta) \cdot \underline{i}_s^s$$

$$\frac{d\hat{\underline{\theta}}}{dt} = \mathbf{L}_c \cdot \underline{\varepsilon}_s^r(t,\hat{\underline{\theta}}) \qquad \underline{\varepsilon}_s^r = \left[\varepsilon_d(t,\hat{\underline{\theta}}) \quad \varepsilon_q(t,\hat{\underline{\theta}})\right]^T \qquad (5)$$

$$\mathbf{\Psi}^T = -\frac{d\underline{\varepsilon}_s^r(t,\hat{\underline{\theta}})}{d\hat{\underline{\theta}}} = \frac{d\hat{\underline{i}}_s^r(t,\hat{\underline{\theta}})}{d\hat{\underline{\theta}}} = \left[\frac{d\hat{\underline{i}}_s^r(t,\hat{\psi}_m)}{d\hat{\psi}_m} \quad \frac{d\hat{\underline{i}}_s^r(t,\hat{r}_s)}{d\hat{r}_s}\right]$$

The values of the prediction errors for 10% under-estimated ψ_m is shown in Fig. 3. It is evident that the δi_d ($= \underline{\varepsilon}_d$) is more consistently sensitive to incorrect ψ_m than its q-axis counterpart (δi_q), therefore, it will contribute in ψ_m adaptation across the whole speed range, except at zero speed, at which, the prediction error goes to zero. On the contrary, the δi_d, δi_q in Fig. 4. (a) and (b) do not show distinct differences despite r_s underestimation, therefore, it is interesting to investigate their individual contributions for online r_s adaptation. However, it is immediately evident that, unlike in the case of ψ_m, the δi_d and δi_q for r_s are dominant only around zero speed. This hints us that the adaptation of ψ_m and r_s can be conveniently decoupled in different speed ranges even though $\underline{\varepsilon}_s$ contains information about deviations of both parameters at nonzero speeds. The steady-state prediction errors when both parameters contain deviations from their estimated values are given in (6) which corroborates the plots in Fig. 3 & 4. It also tells that $\underline{\varepsilon}_s$ is load dependent.

$$\varepsilon_d = -\frac{n^2 \cdot \hat{x}_q}{\hat{r}_s^2 + n^2 \cdot \hat{x}_q \cdot \hat{x}_d} \cdot (\psi_m - \hat{\psi}_m) - \left[\frac{\hat{r}_s}{\hat{r}_s^2 + n^2 \cdot \hat{x}_q \cdot \hat{x}_d} \cdot i_d + \frac{n \cdot \hat{x}_q}{\hat{r}_s^2 + n^2 \cdot \hat{x}_q \cdot \hat{x}_d} \cdot i_q\right] \cdot (r_s - \hat{r}_s)$$

$$\varepsilon_q = -\frac{n \cdot \hat{r}_s}{\hat{r}_s^2 + n^2 \cdot \hat{x}_q \cdot \hat{x}_d} \cdot (\psi_m - \hat{\psi}_m) - \left[\frac{\hat{r}_s}{\hat{r}_s^2 + n^2 \cdot \hat{x}_q \cdot \hat{x}_d} \cdot i_q - \frac{n \cdot \hat{x}_d}{\hat{r}_s^2 + n^2 \cdot \hat{x}_q \cdot \hat{x}_d} \cdot i_d\right] \cdot (r_s - \hat{r}_s) \qquad (6)$$

$$\varepsilon_d = -\left[\frac{i_d}{\hat{r}_s}\right] \cdot (r_s - \hat{r}_s) = -i_d \cdot \left(\frac{r_s}{\hat{r}_s} - 1\right)$$

$$\varepsilon_q = -\left[\frac{i_q}{\hat{r}_s}\right] \cdot (r_s - \hat{r}_s) = -i_q \cdot \left(\frac{r_s}{\hat{r}_s} - 1\right) \qquad (7)$$

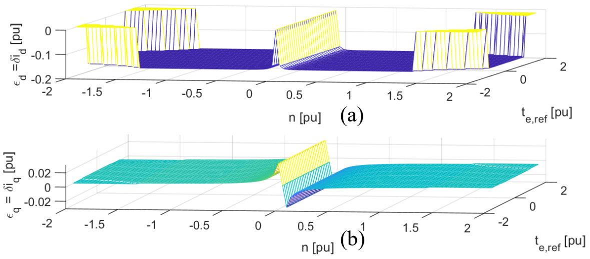

Fig. 3. Sensitivity plot of prediction errors w.r.t. ψ_m (a) d-axis prediction error (b) q-axis prediction error

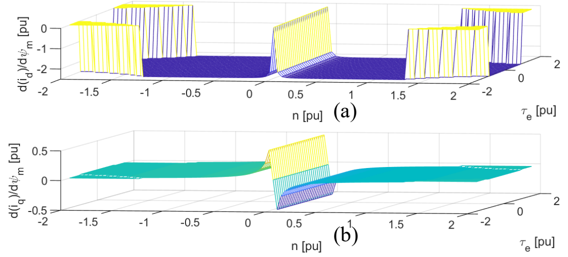

Fig. 5. Gradients of prediction errors w.r.t. ψ_m for (a) d-axis gradient (b) q-axis gradient

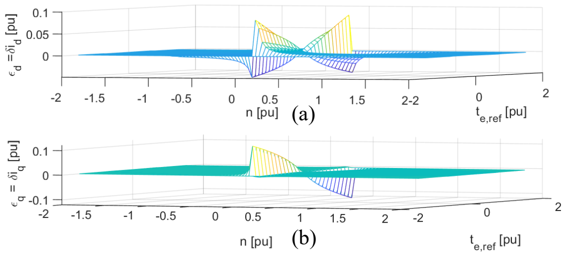

Fig. 4. Sensitivity plot of prediction errors w.r.t. r_s (a) d-axis prediction error (b) q-axis prediction error

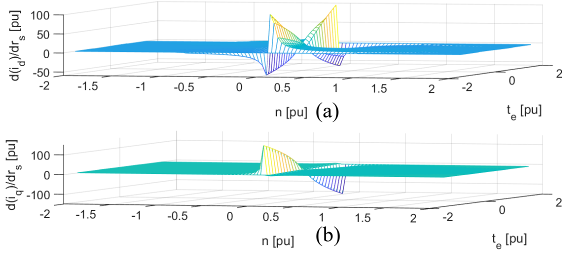

Fig. 6. Gradients of prediction errors w.r.t. r_s for (a) d-axis gradient (b) q-axis gradient

At standstill, the prediction errors become as in (7). Accordingly, depending on the magnitude of i_d and i_q at standstill, r_s can be identified. This provides also an experimental basis for r_s identification at the start-up of the machine.

V. Prediction-Error Gradients Analysis

A. For PM Flux Linkage Estimate

When ψ_m is estimated, the dynamic model of the PEGs, i.e. $\mathbf{\Psi}^T$ becomes, in component form:

$$
\begin{aligned}
\frac{d\left(\dfrac{d\hat{i}_d}{d\hat{\psi}_m}\right)}{dt} &= -\frac{1}{T_d}\cdot\frac{d\hat{i}_d}{d\hat{\psi}_m} + \frac{\omega_n\cdot n\cdot\hat{x}_q}{\hat{x}_d}\cdot\frac{d\hat{i}_q}{d\hat{\psi}_m} \\
\frac{d\left(\dfrac{d\hat{i}_q}{d\hat{\psi}_m}\right)}{dt} &= -\frac{1}{T_q}\cdot\frac{d\hat{i}_q}{d\hat{\psi}_m} - \frac{\omega_n\cdot n\cdot\hat{x}_d}{\hat{x}_q}\cdot\frac{d\hat{i}_d}{d\hat{\psi}_m} - \frac{\omega_n\cdot n}{\hat{x}_q}
\end{aligned}
\tag{8}
$$

This model has the same eigenvalues as the model $\mathcal{M}_{u\vartheta}$, therefore it can be assumed stable. The dynamics of the PEGs are given by d- and q-axis time constants, T_d, T_q and n which is also the input or excitation for this dynamic system. The steady state solutions of these equations are:

$$
\begin{aligned}
\frac{d\hat{i}_d}{d\hat{\psi}_m} &= -\frac{n^2\cdot\hat{x}_q}{\hat{r}_s^2 + n^2\cdot\hat{x}_q\cdot\hat{x}_d} \\
\frac{d\hat{i}_q}{d\hat{\psi}_m} &= -\frac{n\cdot\hat{r}_s}{\hat{r}_s^2 + n^2\cdot\hat{x}_q\cdot\hat{x}_d}
\end{aligned}
\tag{9}
$$

The gradient of the d-axis prediction error becomes $-1/x_d$ in most of the speed range and is independent of torque. The q-axis component becomes quite small due to r_s -dependency. Both functions become zero at standstill. These functions are plotted in the torque-speed plane in Fig. 5. From these plots it can be inferred that the d-component of the prediction error should be used for estimation of ψ_m. When implementing the model in a digital controller, the method of discretization must be considered as well. Usually the Forward Euler Method is numerically accurate enough. It is then important to investigate the stability of the discrete model for both $\mathcal{M}_{u\vartheta}$ and $\mathbf{\Psi}^T$. This can be done by investigating the locations of the poles in the λ*h-plane, where $h = T_{samp}$.

B. For Stator Resistance Estimate

When r_s is estimated, the dynamic model becomes, in component form:

$$
\begin{aligned}
\frac{d\left(\dfrac{d\hat{i}_d}{d\hat{r}_s}\right)}{dt} &= -\frac{1}{T_d}\cdot\frac{d\hat{i}_d}{d\hat{r}_s} + \frac{\omega_n\cdot n\cdot\hat{x}_q}{\hat{x}_d}\cdot\frac{d\hat{i}_q}{d\hat{r}_s} - \frac{\omega_n}{\hat{x}_d}\cdot i_d \\
\frac{d\left(\dfrac{d\hat{i}_q}{d\hat{r}_s}\right)}{dt} &= -\frac{1}{T_q}\cdot\frac{d\hat{i}_q}{d\hat{r}_s} - \frac{\omega_n\cdot n\cdot\hat{x}_d}{\hat{x}_q}\cdot\frac{d\hat{i}_d}{d\hat{r}_s} - \frac{\omega_n}{\hat{x}_q}\cdot i_q
\end{aligned}
\tag{10}
$$

The steady state solution is plotted in Fig. 6 and becomes:

$$
\begin{aligned}
\frac{d\hat{i}_d}{d\hat{r}_s} &= -\frac{\hat{r}_s}{\hat{r}_s^2 + n^2\cdot\hat{x}_q\cdot\hat{x}_d}\cdot i_d - \frac{n\cdot\hat{x}_q}{\hat{r}_s^2 + n^2\cdot\hat{x}_q\cdot\hat{x}_d}\cdot i_q \\
\frac{d\hat{i}_q}{d\hat{r}_s} &= -\frac{\hat{r}_s}{\hat{r}_s^2 + n^2\cdot\hat{x}_q\cdot\hat{x}_d}\cdot i_q + \frac{n\cdot\hat{x}_d}{\hat{r}_s^2 + n^2\cdot\hat{x}_q\cdot\hat{x}_d}\cdot i_d
\end{aligned}
\tag{11}
$$

Before concluding this section, it is worth highlighting that the plots of gradient functions irrespective of the parameter, hold the identical shape of their respective prediction error sensitivity plots.

VI. Gain Matrices Computation

The parameter adaptation algorithm in discrete form based on the Forward Euler Method, which becomes:

$$
\underline{\hat{\theta}}[k] = \left[\underline{\hat{\theta}}[k-1] + \mathbf{L}\cdot\underline{\varepsilon}_s^r[k]\right]_{D_M} \tag{12}
$$

where $D_{\mathcal{M}}$ is in the stable region of the model $D_{\mathcal{M}} \subset D_s$. This means that all model parameters and sampling time

T_{samp} must be chosen such that the discrete model is stable. The parameter space for the model is limited to:

$$D_M = \left\{ \begin{matrix} \psi_{m,\min} & \le & \hat{\psi}_m & \le & \psi_{m,\max} \\ r_{s,\min} & \le & \hat{r}_s & \le & r_{s,\max} \end{matrix} \right\} \tag{13}$$

A. Stochastic Gradient Algorithm

General *stochastic gradient algorithm* (SGA), as per [8], can be expressed as:

$$\begin{aligned} &\mathbf{L} = \gamma[k]\cdot\frac{\mathbf{\Psi}[k]}{r[k]} \\ &r[k] = r[k-1] + \gamma[k]\cdot\left\{tr\left[\mathbf{\Psi}[k]\cdot\mathbf{\Psi}^T[k]\right] - r[k-1]\right\} \end{aligned} \tag{14}$$

Here, r[k] is the scalar version of the Hessian matrix used in this algorithm and the trace (*tr*) of a matrix is the sum of the diagonal elements. The gain-sequence γ could be time dependent, but a constant value γ_0 is usually chosen. This memory coefficient γ_0 of the algorithm should be chosen such that the parameter is "almost constant" within the time period $T_0 = T_{samp}/\gamma_0$ [3]. The initial value of r[k] in the 1st order filter in (13) help boost the gain $\mathbf{L}$ during start-up. It is also possible to choose a different γ[k] in the filter for r[k] and the gain $\mathbf{L}$. The PEG $\mathbf{\Psi}^T$ and the trace of $\mathbf{\Psi}\,\mathbf{\Psi}^T$ can be expressed as:

$$\begin{aligned} &\mathbf{\Psi}^T[k] = \begin{bmatrix} \dfrac{d\hat{i}_d}{d\hat{\psi}_m}[k] & \dfrac{d\hat{i}_d}{d\hat{r}_s}[k] \\ \dfrac{d\hat{i}_q}{d\hat{\psi}_m}[k] & \dfrac{d\hat{i}_q}{d\hat{r}_s}[k] \end{bmatrix} \\ &tr\left\{\mathbf{\Psi}[k]\cdot\mathbf{\Psi}^T[k]\right\} = \left(\frac{d\hat{i}_d}{d\hat{\psi}_m}\right)^2 + \left(\frac{d\hat{i}_q}{d\hat{\psi}_m}\right)^2 + \left(\frac{d\hat{i}_d}{d\hat{r}_s}\right)^2 + \left(\frac{d\hat{i}_q}{d\hat{r}_s}\right)^2 \end{aligned} \tag{15}$$

The generalized SGA makes use of the dynamic model of the PEGs [8] as well as a common scalar value r[k], which is the filtering of the trace of $\mathbf{\Psi}\,\mathbf{\Psi}^T$. However, when IPMSM is concerned, it is interesting to investigate both steady and dynamic states of PEGs in the SGA. By inspection of (8) and (10), both these 2nd order systems share the eigenvalues of $\mathcal{M}_{u9}$ given in (4). This means that similar oscillations can occur in the PEGs and thus the gain matrix $\mathbf{L}$ during transient operations. Furthermore, (8) tells that the dynamic model of the PEGs w.r.t. ψ_m is excited by the speed only. While the speed usually has a low derivative due to the inertia of the system, it is sensible to use the steady state (std) solution of the PEGs over their dynamic (dyn) counterparts in computing the respective gains, L_{11} and L_{12}. On the other hand, the model for the PEGs w.r.t. r_s are excited by the currents (at and around zero speed) as in (10). These currents can change very rapidly, such that the dynamic PEGs should be applied to obtain some sort of a filtering effect when calculating L_{21} and L_{22}. Close inspection of (10) will tell that, at standstill, these PEGs are decoupled and with filter time-constants T_d and T_q. Thus, it is logical to select the dynamic PEGs when r_s adaptation is concerned. However, it is interesting to observe how steady state PEGs will contribute in r[k] calculation.

The next step is to determine which PEGs to be employed in the trace in r[k] calculation and whether the trace should be filtered or not. Between the choice of filtered versus unfiltered r[k], the first becomes an obvious choice as to make the outcome free from oscillations which would have otherwise been superimposed on parameter-estimation trajectory. The filtered r[k]-variants in (16) become the promising alternatives, of which the effects are seen in section VII. The dynamics of the filtered value of r[k] expressions in (16) are plotted in Fig. 7. It must be noted that r_n[k] and $r_{\psi m}$[k] are plotted for acceleration-cases from standstill, while the remaining are for cases at standstill. It is seen that $r_{\psi m}$[k] is maintained at a very low value opposing to $r_{rs1,2}$[k]. PEGs in r_n[k] is the sum of these two extreme cases, thus it initially takes off, but as speed increases, the effect of r_s -PEGs becomes insignificant.

$$\begin{aligned} &r_n[k]:\ tr\left\{\mathbf{\Psi}[k]\cdot\mathbf{\Psi}^T[k]\right\} = \left(\frac{d\hat{i}_d}{d\hat{\psi}_m}\right)^2_{std} + \left(\frac{d\hat{i}_q}{d\hat{\psi}_m}\right)^2_{std} + \left(\frac{d\hat{i}_d}{d\hat{r}_s}\right)^2_{dyn} + \left(\frac{d\hat{i}_q}{d\hat{r}_s}\right)^2_{dyn} \\ &r_{\psi m}[k]:\ tr\left\{\mathbf{\Psi}[k]\cdot\mathbf{\Psi}^T[k]\right\} = \left(\frac{d\hat{i}_d}{d\hat{\psi}_m}\right)^2_{std} \\ &r_{rs1}[k]:\ tr\left\{\mathbf{\Psi}[k]\cdot\mathbf{\Psi}^T[k]\right\} = \left(\frac{d\hat{i}_d}{d\hat{r}_s}\right)^2_{dyn} + \left(\frac{d\hat{i}_q}{d\hat{r}_s}\right)^2_{dyn} \\ &r_{rs2}[k]:\ tr\left\{\mathbf{\Psi}[k]\cdot\mathbf{\Psi}^T[k]\right\} = \left(\frac{d\hat{i}_d}{d\hat{r}_s}\right)^2_{std} + \left(\frac{d\hat{i}_q}{d\hat{r}_s}\right)^2_{std} \end{aligned} \tag{16}$$

Therefore, r_n[k] eventually converges with $r_{\psi m}$[k]. Since r[k] is in the denominator in the $\mathbf{L}$-computation formula in (14), relatively smaller r[k] values are expected to amplify $\mathbf{L}$, thus rate of convergence. From this perspective, it can be predicted that $r_{\psi m}$[k] will offer faster rate of convergence than r_n[k] when ψ_m adaptation is concerned. It is also evident that between r_{rs1}[k] and r_{rs2}[k], the earlier follows a lower trajectory in the beginning, which is favorable as per the algorithm. Due to this reason and the advantages of dynamic r_s-PEGs discussed earlier, it is fair to disregard steady state PEGs for r_s and r_{rs2}[k] in the subsequent discussions. On a side note, the unfiltered values of r[k] are equal to the steady state values of the traces in Fig. 7.

Consequently, the three versions of these algorithms for $\mathbf{L}$-computation are of special interest:

- Filtered r_n[k] and corresponding dynamic $\mathbf{\Psi}^T$
- Filtered $r_{\psi m}$[k] and corresponding steady state $\mathbf{\Psi}^T$
- Filtered r_{rs1}[k] and corresponding dynamic $\mathbf{\Psi}^T$

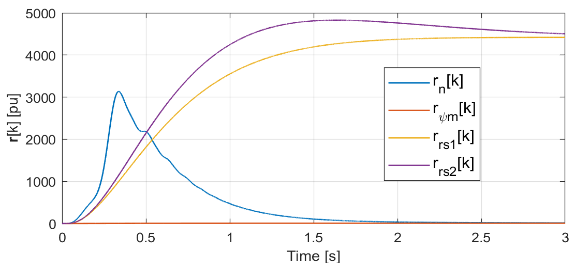

Fig. 7. r[k] function behavior when constructed with different trace combinations corresponding to (16)

If only ψ_m is incorrect, for the steady state $-\mathbf{\Psi}^T$ and unfiltered $r_{\psi m}[k]$, one obtains:

$$L_{11} = -\gamma_0 \cdot \left(\frac{\hat{r}_s^2}{n^2[k] \cdot \hat{x}_q} + \hat{x}_d \right) \quad |n| > n_{\lim} \tag{17}$$

$$L_{11} \cdot \varepsilon_d = \gamma_0 \cdot (\psi_m - \hat{\psi}_m)$$

This corresponds to the gain chosen by only interpreting the steady-state prediction error in (6). It is important to limit the gain L_{11} at low speeds to avoid amplification of the noise in the current measurements.

VII. Simulation Results & Discussion

A 3-phase IPMSM drive with a 2-level inverter and different loads has been simulated in MATLAB Simulink/Simscape toolbox. A constant load at zero speed has been used for r_s online-estimation and a quadratic load for the case of ψ_m. Asymmetrical modulation with 3rd harmonic injection has been used. The switching frequency is 3 kHz and the sampling frequency of the controller is 6 kHz. In these simulations, the PAA is started immediately at start-up of the drive. TABLE I contains the simulation data.

TABLE I. Simulation Data

	Symbol	***Value***	***Unit***
Nominal voltage	U_N	690	V
Nominal current	I_N	478	A
Nominal frequency	f_N	50	Hz
Pole pairs	p	1	-
Motor parameter vector	$[\psi_m\ x_d\ x_q\ r_s]^T$	$[0.66\ 0.4\ 1\ 0.009]^T$	pu
Initial estimated parameter vector	$[\hat{\psi}_m\ \hat{x}_d\ \hat{x}_q\ \hat{r}_s]^T$	$[0.59\ 0.4\ 1\ 0.008]^T$	pu

A. Online Adaptation of PM Flux Linkage

It has been earlier revealed that ε_d was the component most sensitive to ψ_m. Based on that, when only ψ_m is adapted beyond the low speed range, a simpler algorithm can be derived. With this aim, simulations were performed with different L-combinations. It is obvious that, among the r[k] expressions in (16), the choices are limited to the first two expressions as per SGA. Fig. 8 presents the results. Due to the reasons explained in connection to Fig. 7, relatively large $r_n[k]$ impedes the ψ_m adaptation briefly. This aspect is further consolidated when, despite L_{21} and L_{22} are disabled, the convergence speed remains nearly the same, as long as the same r[k] is employed across all gain-combinations. Alternatively, when $r_{\psi m}[k]$ is used, ψ_m adaptation becomes rapid to reinforce the arguments in connection to Fig. 7. Despite the rate of convergence is adjustable by tuning the γ_0, and a convergence within a few seconds is sufficient in this context. This investigation has revealed that it is the use of L_{11} only and $r_{\psi m}[k]$ that offers the fastest natural convergence for ψ_m in terms of stochastic gradient algorithm.

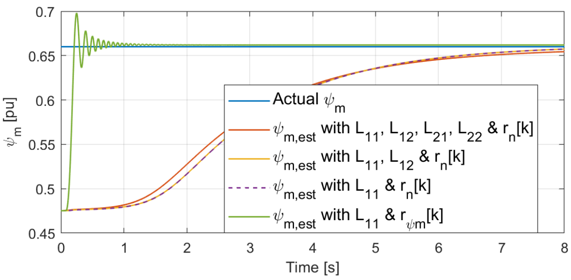

Fig. 8. Online PM flux linkage estimation with different gains and Hessian

B. Experimentation of Stator Resistance at Start-up

To simulate a start-up scenario of an industrial drive, the simulation model was run by decoupling the controller from MTPA strategy and feeding in following references; $i_q = 0$, $i_d = 0.5$, $\vartheta_{rotor} = 0$. Such setting results in a motor that receives magnetizing current in its physical a-winding and operates at zero speed, creating zero torque. As it is evident from (9), the PEGs of ψ_m become zero at standstill, thus L_{11} and L_{12} become zero as well as no contribution in the r[k] computation. This means that it is fair to apply $r_{rs1}[k]$ in the SGA. Furthermore, since i_q is kept zero, PEG of r_s q-component become zero at zero-speed (10), therefore, L_{22} can be neglected in the PAA. It is effectively the d-component of r_s PEG and its corresponding gain L_{21} is in use in this experiment. Fig. 9 illustrates the performance comparisons.

Despite using the d-component of the r_s-PEG alone is enough in the respective trace to estimate r_s, including both L_{21} and L_{22} as well as both PEGs for r_s in calculating the trace makes the algorithm more applicable for other input currents as well. This will be clearer in the next section.

C. Online Adaptation of Stator Resistance

In this simulation, we return to the MTPA strategy and supplement with a speed controller to maintain zero rotational speed while obtaining constant torque to serve a constant load. Reference currents: $i_q = 0.5$, $i_d = -0.33$. As said before, PEGs of ψ_m do not contribute at zero-speed, thus it is logical to apply $r_{rs1}[k]$ in **L**-computation algorithms, and disregard L_{11} and L_{12} in the PAA. Different permutations of L_{21} and L_{22} were investigated (see Fig. 10) and it turned out that the best case is when both these gains are combined, while both currents contribute to the gains and prediction errors as shown in (7) and (11).

D. Simultaneous Adaptation & Gain Scheduling

The prediction errors ε_d and ε_q contain information about both ψ_m and r_s errors. Refer (4). Assuming, the drive starts from standstill, when both parameters are attempted to adapt simultaneously, the smaller parameter r_s gets

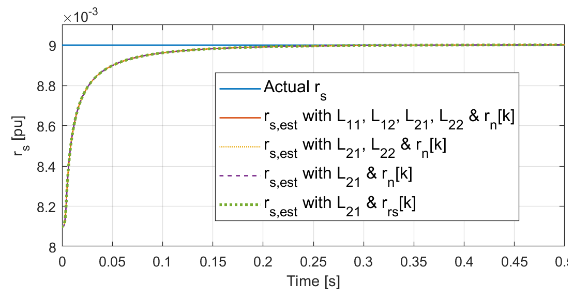

Fig. 9. Stator resistance estimation at start-up with different gains and Hessian

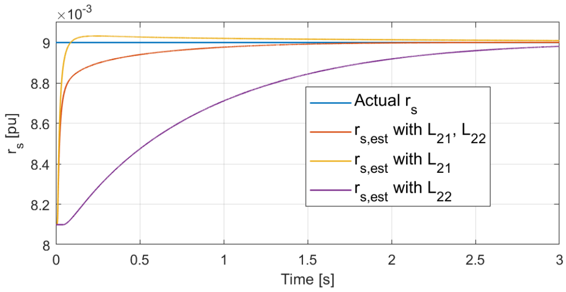

Fig. 10. Contribution from different gains and their combinations in online r_s estimation

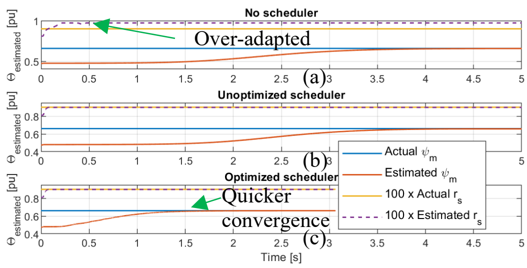

Fig. 11. Parameter convergence with the gain-scheduling mechanism (a) no scheduler (b) with scheduler (c) with optimized scheduler

compensated even due to ψ_m -parameter error and saturated in its upper limit in the parameter space as shown in Fig. 11(a). After r_s gets wrongly settled, while the rotor-speed is increasing, the PAA attempts to minimize the remaining prediction-errors by adapting ψ_m. However, this challenge can be conveniently circumvented by adapting only one parameter at a time by scheduling $\mathbf{L}$ in different speed regions.

When FE-based position estimation is concerned, r_s adaptation becomes necessary at and around zero speed, where ε-sensitivity for ψ_m is weak (see Fig. 3). Therefore, it is justifiable to disable r_s adaptation beyond very-low speeds (n) and perform ψ_m estimation alone in the remaining speed range. With this aim, a scheduling mechanism has been introduced in the simulation in which at $n_{lim} > 0.01$ [pu], r_s adaptation has been disabled. Besides, the SGA-settings are identical to the case previous case of not having a scheduler at all ($r[k] = r_n[k]$ and corresponding $\mathbf{\Psi}^T$ in $\mathbf{L}$-computation). The r_s estimation shown Fig. 11(b) is more accurate than the case before. The gain-scheduling can be optimized by additionally disabling the unused traces in r[k] calculation, such that when $n_{lim} > 0.01$ [pu], $r_n[k] = r_{\psi m}[k]$. Performance of such scheme is given in the Fig. 11(c), where ψ_m converges much earlier.

VIII. Conclusion

This paper proposed an effective and flexible method for parameter adaptation using prediction-error gradients to adapt motor-parameters online. The method was presented step-by-step to estimate stator resistance and the PM flux linkage in an IPMSM drive in order to establish a general framework for online parameter identification of any electric drive. The use of prediction-error gradient functions is more effective when its steady-state solution is used for ψ_m and dynamic counterpart is used for r_s -adaptations. The stochastic gradient algorithm that is applied to compute the gain-matrices offers a range of variables to tune the convergence performance. Among which, the scalar Hessian function plays a pivotal role. Stator resistance was adapted in the zero and very low speed range, as required and PM flux linkage was adapted in the remaining speed range. To decouple the parameter adaptation without compromising the accuracy, an optimized gain-scheduling mechanism was proposed. Investigation of time dependent and optimized gain-sequences can be an interesting future research work.

References

[1] A. Piippo, M. Hinkkanen, and J. Luomi, "Adaptation of motor parameters in sensorless PMSM drives," *IEEE Trans. Ind. Appl.*, vol. 45, no. 1, pp. 203–212, 2009.

[2] H. Ahn, H. Park, C. Kim, and H. Lee, "A Review of State-of-the-art Techniques for PMSM Parameter Identification," *J. Electr. Eng. Technol.*, vol. 15, no. 3, pp. 1177–1187, 2020.

[3] X. Li and R. Kennel, "Comparison of state-of-the-art estimators for electrical parameter identification of PMSM," in *PRECEDE 2019: 2019 IEEE International Symposium on Predictive Control of Electrical Drives and Power Electronics*, 2019.

[4] A. Khlaief, M. Boussak, and A. Châari, "A MRAS-based stator resistance and speed estimation for sensorless vector controlled IPMSM drive," *Electric Power Systems Research*, vol. 108. pp. 1–15, 2014.

[5] R. Van Der Merwe and E. A. Wan, "The square-root unscented Kalman filter for state and parameter-estimation," *ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc.*, vol. 6, pp. 3461–3464, 2001.

[6] 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 Trans. Ind. Appl.*, vol. 47, no. 2, pp. 798–804, 2011.

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

[8] L. Ljung and T. Soderstrom, *Theory and Practice of Recursive Identification*, 2nd Editio. Cambridge, Massachusetts: The MIT Press, 1985.

[9] R. Nilsen and M. P. Kazmierkowski, "Reduced-order observer with parameter adaption for fast rotor flux estimation in induction machines," vol. 136, no. I, pp. 35–43, 1989.

[10] A. Perera and R. Nilsen, "A Sensorless Control Method for IPMSM with an Open-Loop Predictor for Online Parameter Identification," in *23rd International Conference on Electrical Machines and Systems, ICEMS, In Press*, 2020, p. 6.

[11] A. Perera and R. Nilsen, "A Method Based on Prediction-Error-Gradients to Estimate PMSM Parameters Online," in *Conference Record - IAS Annual Meeting (IEEE Industry Applications Society)*, *In Press*, 2020, pp. 2–8.