syreDrive: Automated Sensorless Control Code Generation for Synchronous Reluctance Motor Drives

syreDrive: Automated Sensorless Control Code Generation for Synchronous Reluctance Motor Drives

Anantaram Varatharajan, Dario Brunelli, Simone Ferrari, Paolo Pescetto and Gianmario Pellegrino
Department of Energy
Politecnico di Torino
Turin, Italy
anantaram.varatharajan@polito.it

***Abstract*—To compete with scalar controlled induction motors, Synchronous Reluctance (SyR) motor drives for general purpose applications must be sensorless controlled, which is considered a non-standard practice in the field of electrical drives control. This paper proposes a new software tool for the simulation of a sensorless controlled SyR machine drive, with automatic generation of the control code and the simulation model itself, starting from the motor flux maps and main design parameters. This tool is integrated in the motor design platform SyRe to bridge the motor desing and control design phases into a single, integrated toolchain. Results are shown for two motors of different ratings, to assess the general validity of the proposed approach.**

***Index Terms*—Sensorless control, synchronous reluctance machine, automated simulation model.**

I. Introduction

Synchronous Reluctance (SyR) machines are the higher efficiency alternative to variable speed induction motors for general purpose applications and has received increasing research and commercial interest since the past decade. The saliency of SyR machines aid in reliable sensorless position estimation that further tips the scale in its favor over induction machines for variable speed applications.

Strides of progress has been made in the SyR machine design towards improving saliency and torque density. Equally so in the field of commissioning and control, considering the nonlinear magnetic model of the SyR machines. A shared expertise in the fields of design and control bring synergy to the overall integrated system. This work is in the spirit of bridging the machine design and drive control arenas by introducing a tool for automated control simulation generation within the motor design environment. In particular, the evaluation of sensorless control capability finds importance for SyR machines which usually do not have a position transducer off-the-shelf.

The main contributions of this work are enumerated as follows:

1) The motor design framework of SyR-e is supplemented with motor drive simulation feature that allows for a quick and streamlined control performance evaluation of design prototypes.
2) A machine-independent general template for sensorless control is developed that automatically calibrates using the flux maps of the machine under test. This provides for an initial starting point which the user can further optimize should he/she desire it.
3) The sensorless control algorithm is comprehensive with signal-injection based observers at low speeds region and fundamental-wave (back-emf) based observers at high speeds region. They are fused in a smooth speed-dependent linear fashion.

This work was supported by Power Electronics and Innovation Center (PEIC) of Politecnico di Torino.

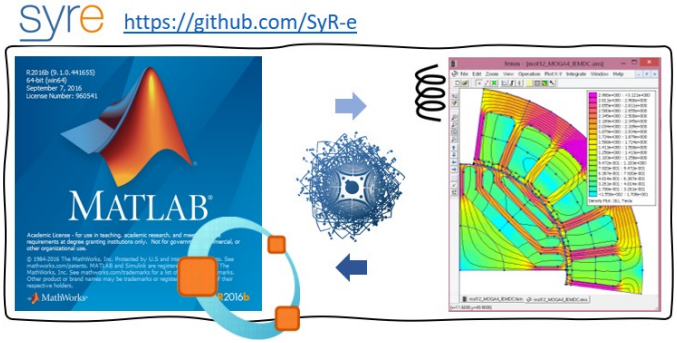

Fig. 1. The working principle behind SyR-e [1] that relies on FEMM [2] and Matlab [3] environments.

Section II introduces SyR-e and describes the automated simulation generation feature of syreDrive. Section III discusses the portfolio of various sensorless control techniques available to the user. Section IV presents simulation results for two motors of different sizes. Finally, Section V concludes the paper.

II. Description of syreDrive

As a precursor to syreDrive, the open-source motor design tool SyR-e is first briefly introduced.

A. SyR-e: An Open-Source Motor Design Tool

SyR-e [1] stands for Synchronous Reluctance evolution and is an open-source framework for electrical machine design and FEA analysis. The basic principle of SyR-e is explained in Fig. 1: design and simulation routines implemented in Matlab [3] setup using FEA models in FEMM [2]. Simulations are performed in FEMM and the results are automatically streamlined to Matlab for further post-processing and elaborations. A simple Graphical User Interface (GUI) completes the Matlab environment, simplifying the design and simulation workflow and making the tool usable beyond the niche research community.

Historically, SyR-e was first released in 2014 for SyR machines design through multi-objective optimization algorithm. Thereafter, it evolved over the years with new geometries (including PM machines), preliminary design models, migration to other FEA tools and a comprehensive environment for FEA analysis and validation. The latter section includes several kind of analysis, from the simple evaluation of the motor outputs for a single operating point, to analysis on characteristic current and demagnetization of PM machines and flux maps evaluation.

SyR-e embeds some functions for the flux maps elaboration to compute optimal control trajectories and operating limits, in addition to the efficiency maps. These features are grouped in a second GUI, identified with the acronym MMM, meaning Magnetic Model Manipulation. Besides the data processing reported before, it also contains the information to build a dynamic model of the motor. This is done with the new feature called "syreDrive", that automatically creates the dynamic model of the motor in Simulink, including the data of the drive and the adopted control strategy. This is the subject of the following sections.

B. syreDrive: Automated Simulink Model Assembly

As alluded to earlier, syreDrive is a tool that automatically adapts a general template of the control simulation model to the considered prototype following the machine design tool-chain. Thanks to the automated generation, the user's interaction is solely limited to the graphical user interface (GUI) shown in Fig. 2 and does not need to tackle any matters related to the control, aside from choosing the type of control algorithm among a range of options. The syreDrive tab within the MMI GUI allows the user to customize the control type between speed, torque and current control, and the flux maps model between dq (2D) and $dq\theta$ (3D), discussed in the following section. The options related to the sensorless control are dealt with in Section III.

The digital control in the Simulink model is structured using C-files in the form of S-function to be compatible for easy migration to microcontroller. The flow chart of the syreDrive is shown in Fig. 3. Once the *RUN* button is triggered, a header file with motor parameters, flux maps and the optimal control trajectories, i.e., maximum torque per ampere (MTPA) and maximum torque per volts (MTPV), corresponding to the motor under inspection is generated. This is then supplemented

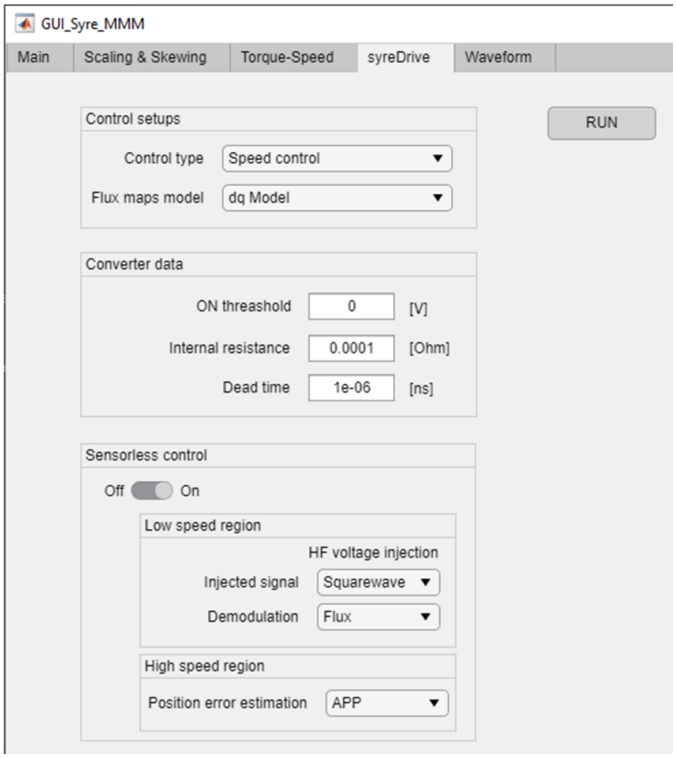

Fig. 2. Graphical user interface of syreDrive within the SyR-e magnetic model manipulation framework.

to the general simulation model template together with the user input settings to build the customized model. The simulation is run for a predefined reference sequence and the results are printed at the completion.

C. Efficient 3D Interpolation Technique

Flux maps are one of the most important and complete ways to express the magnetic model of SyR machines. SyR-e can handle two kind of flux maps: dq and $dq\theta$ models. The former expresses the dq flux linkages as functions of the dq currents. The model is simple, but spacial harmonic effects are neglected (the map is averaged over one electrical period). The latter is more complex and expresses flux linkages and torque as functions of the dq currents and the rotor position θ. This augmented complexity is balanced from the capability to model also harmonic effects and torque ripple evaluation.

The machine model automatically designed by syreDrive is structured with a computationally efficient algorithm based on a two steps interpolation. The model is capable of alternatively manage the dq or $dq\theta$ model according to the user settings. In both cases, the stator flux is obtained by integrating the motor electromagnetic force, and the current is retrieved by the inverse magnetic model of the machine (i.e. $\boldsymbol{i}_{dq}$ characteristic as a function of $\boldsymbol{\lambda}_{dq}$) computed by SyR-e. If the dq model is selected, the inverse magnetic model is implemented in form of 2D LUTs, while the torque is computed with (3). Conversely, in the $dq\theta$ model both the instantaneous currents

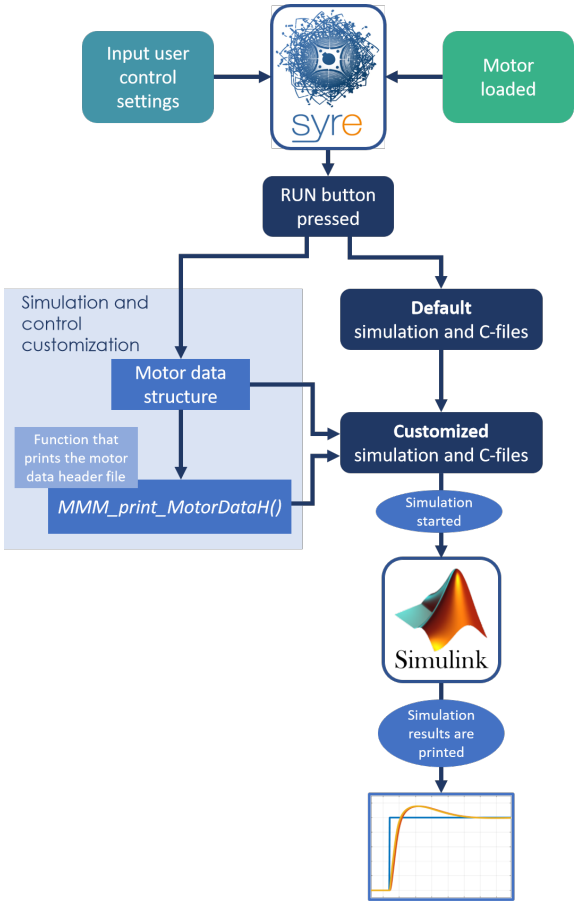

Fig. 3. Flowchart of syreDrive for automated simulation model assembly for quick machine design evaluations.

and torque are extracted by 3D LUTs, which include the dependency on rotor position.

Thanks to the efficiency of the implemented interpolation algorithm and SyR-e data organization, the additional computational burden required by the $dq\theta$ model is limited, and the simulation time increases only about 10% respect to the dq case.

III. Summary of Sensorless Control Techniques

A. Mathematical Model of a Synchronous Reluctance Machine

The electrical rotor position is θ and the electrical angular speed is $\omega = s\,\theta$ where s is the differential operator $\frac{\mathrm{d}}{\mathrm{d}t}$. The orthogonal rotational matrix is $\mathbf{J} = \left[\begin{smallmatrix} 0 & -1 \\ 1 & 0 \end{smallmatrix}\right]$ and $\mathbf{I}$ is the identity matrix. Real space vectors will be used; for example, the stator current is $\boldsymbol{i}_{dq} = [i_d, i_q]^{\mathrm{T}}$ where i_d and i_q are the vector components in the dq rotor reference frame. Space vectors in the stationary reference frame are denoted by subscript $\alpha\beta$.

The voltage equation of a synchronous machine in the dq rotor reference frame is given by

$$s\,\boldsymbol{\lambda}_{dq} = \boldsymbol{v}_{dq} - R_s \boldsymbol{i}_{dq} - \omega\,\mathbf{J}\,\boldsymbol{\lambda}_{dq} \tag{1}$$

where R_s is the stator resistance and $\boldsymbol{\lambda}_{dq}$ is the stator flux linkage. The linearized small-signal form of the magnetic model can be expressed as

$$\frac{\partial \boldsymbol{\lambda}_{dq}}{\partial \boldsymbol{i}_{dq}} = \boldsymbol{L}_{\partial}(\boldsymbol{i}_{dq}) = \begin{bmatrix} l_d & l_{dq} \\ l_{dq} & l_q \end{bmatrix} \tag{2}$$

where $\boldsymbol{L}_{\partial}$ is the incremental inductance matrix, l_d, l_q represent the inductance along direct d and quadrature q axis, respectively, while l_{dq} is the cross-saturation term. The inductance matrix is a function of $\boldsymbol{i}_{dq}$. The average electromagnetic torque is given by

$$T_\mu = \frac{3p}{2}\,\boldsymbol{i}_{dq}^{\mathrm{T}}\,\mathbf{J}\,\boldsymbol{\lambda}_{dq} \tag{3}$$

where p is the number of pole pairs. The dynamics of mechanical speed and position, denoted by the subscript r, of a synchronous machine is given by

$$J \cdot s\,\omega_r = T - T_L - B\,\omega_r \tag{4}$$

where J is the shaft inertia, T_L is the load torque and B is the friction coefficient.

B. Control System Overview

The block diagram of the control system is shown in Fig. 4 where the different modes of operation such as speed control, torque control and current control are illustrated along with the different sensorless techniques.

1) Current Vector Control: The classical current vector control with feed-forward compensation is used for closed loop torque/speed control. The cascaded linear proportional-integral (PI) regulators structure is shown in Fig. 4.

The gains current regulators are tuned for an over-damped response with a damping factor $\zeta = 1.58$ as

$$k_{pd} = l_d\,\Omega_T \qquad k_{id} = l_d\,\Omega_T^2/10 \tag{5}$$

where k_{pd} and k_{id} are the d-axis proportional and integral gains, respectively and Ω_T is the bandwidth of the current loop. For an uniform bandwidth, the gains are adapted with the operating point dependent incremental inductance l_d. The q-axis gains are calibrated in a similar fashion using l_q.

2) Hybrid Flux Observer: Let $\boldsymbol{\Lambda}_{dq}(\boldsymbol{i}_{dq})$ denote the flux-map LUTs of the machine under test. The hybrid flux observer state equation in the stationary reference frame, shown in Fig. 5, is defined as

$$s\hat{\boldsymbol{\lambda}}_{\alpha\beta} = \boldsymbol{v}_{\alpha\beta} - R_s \boldsymbol{i}_{\alpha\beta} + \boldsymbol{G}\left(\boldsymbol{\lambda}^i_{\alpha\beta} - \hat{\boldsymbol{\lambda}}_{\alpha\beta}\right) \tag{6}$$

where $\boldsymbol{G}$ is a 2×2 gain matrix, $\hat{\boldsymbol{\lambda}}_{\alpha\beta}$ is the observed stator flux and $\boldsymbol{\lambda}^i_{\alpha\beta} = \boldsymbol{\Lambda}_{dq}(\boldsymbol{i}_{dq})$ is the flux-map LUTs-based current-model stator flux.

For a diagonal matrix $\boldsymbol{G} = g\,\mathbf{I}$, the gain g acts as a cross-over frequency that dictates the dominance of the current-model flux estimation on the observed flux for electrical speeds less g rad/s while the voltage-model flux estimation prevails for speed over g rad/s.

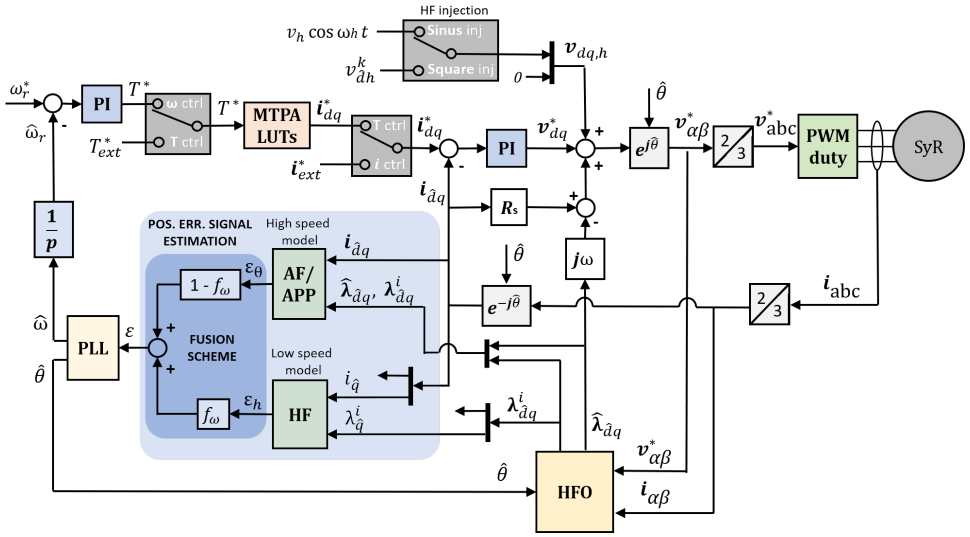

Fig. 4. Block diagram of the control system illustrating the different modes of control (speed/torque/current) and the different sensorless techniques along with the hybrid flux observer and PLL.

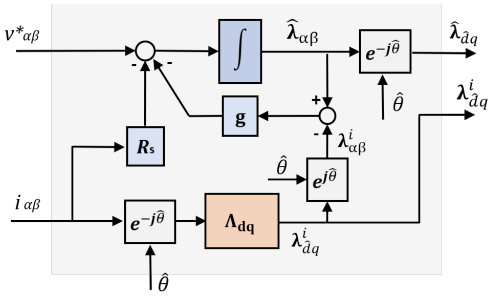

Fig. 5. Block diagram of the hybrid flux observer implemented in the stator reference frame.

C. High-Frequency Excitation-based Low Speed Models

The high-frequency terms are denoted by the subscript h. Among the various forms of signal injection, the pulsating voltage is chosen for the reduced torque ripples. Let the injected sinusoidal voltage magnitude and frequency along estimated $\hat{d}$-axis be denoted as V_h and ω_h, respectively, i.e,

$$\boldsymbol{v}_{\hat{d}qh} = V_h \begin{bmatrix} \cos(\omega_h \mathrm{t}) \\ 0 \end{bmatrix}. \tag{7}$$

A subset of pulsating injection is the square-wave voltage injection at half the switching frequency which is maximum injecting frequency under synchronous sampling. This helps in simplifying the signal processing stage besides providing faster dynamics [4]. Let superscript k denote the discrete domain representation of the k^{th} sampling instant, V_h be the magnitude of injected voltage in $\hat{d}$ axis at half the switching frequency. Then the injected voltage is

$$v_{\hat{d}h}^{k} = V_h \cos(\pi k) = \begin{cases} +V_h, & \text{if } k == 2n \\ -V_h, & \text{if } k == 2n+1 \end{cases} \tag{8}$$

where n is an integer. Note that due to the computational delay, the voltage reference at $k-1$ is the voltage applied at k, i.e., $\boldsymbol{v}_{dq}^{*\ k-1} = \boldsymbol{v}_{dq}^{k}$.

1) q-axis Current Demodulation: The position error signal ϵ_h is designed proportional to the high-frequency current response in the q-axis, $i_{\hat{q}h}$. The high-frequency component $i_{\hat{q}h}$ is extracted from $i_{\hat{q}}$ through a bandpass filter and is demodulated with a heterodyne process to obtain the position error signal ϵ_h as

$$\epsilon_h = -\frac{\omega_h}{2V_h}\, k_\epsilon^i\ \mathrm{LPF}\Big[i_{\hat{q}h} \cdot 2\,\sin(\omega_h t)\Big] \tag{9a}$$

$$k_\epsilon^i = (l_d l_q - l_{dq}^2)/l_\Delta \tag{9b}$$

where k_ϵ^i is a scaling factor [5] and $l_\Delta = (l_d - l_q)/2$. Note that the cut-off frequency of the LPF in (9a) should be atleast three times that of the PLL bandwidth for a healthy phase margin [6]. The equivalent error signal with the square-wave voltage injection is [4]

$$\epsilon_h = -\frac{\cos\Big(\pi(k-1)\Big)}{2V_h}\, k_\epsilon^i\, \frac{i_{\hat{q}}^k - i_{\hat{q}}^{k-1}}{T_s}. \tag{10}$$

where T_s is the sampling interval. Equating $\epsilon_h = 0$ incurs a cross-saturation induced steady-state position error $\tilde{\theta}_{dq}$ given by

$$\tilde{\theta}_{dq} = \frac{1}{2}\tan^{-1}\frac{-l_{dq}}{l_\Delta}. \tag{11}$$

2) q-axis Current-Model Flux Demodulation: The cross-saturation effects can be alleviated by designing the position error signal from the q-axis current-model flux estimate $\lambda_{\hat{q}}^i$ instead of the q-axis current $i_{\hat{q}}$ [7]. The corresponding error signal ϵ_h is obtained for a sinusoidal voltage injection is

$$\epsilon_h = -\frac{\omega_h}{2V_h}\, k_\epsilon^\lambda\ \mathrm{LPF}\Big[\lambda_{\hat{q}h}^i \cdot 2\sin(\omega_h t)\Big]. \tag{12a}$$

$$k_\epsilon^\lambda = (l_d l_q - l_{dq}^2)/(l_q\, l_\Delta - l_{dq}^2) \tag{12b}$$

where the scaling factor is k_ϵ^λ. The equivalent error signal for a square-wave voltage injection is

$$\epsilon_h = -\frac{\cos\Big(\pi(k-1)\Big)}{2V_h}\, k_\epsilon^\lambda\, \frac{\big(\lambda_{\hat{q}}^i\big)^k - \big(\lambda_{\hat{q}}^i\big)^{k-1}}{T_s}. \tag{13}$$

D. Fundamental Excitation-based High Speed Models

At medium to high speeds region, the high-frequency excitation approaches are relegated in favor of the back-emf based fundamental-wave excitation schemes due to their higher reliability and absence of acoustic noise. Two position observer are discussed namely, active-flux observer and adaptive projection vector for position error signal estimation (APP) observer.

1) Active-Flux Position Observer: The active-flux for a SyR machine is defined along the d-axis as an equivalence to the torque produce flux in a non-salient machine as [8]

$$T = \frac{3p}{2}\, \overbrace{(L_d - L_q)\, i_d}^{\boldsymbol{\lambda}_{dq}^{af}}\, i_q \tag{14}$$

where $\boldsymbol{\lambda}_{dq}^{af}$ is the active-flux.

As the active-flux is naturally along the d-axis, the position error signal can be designed proportional to the $\hat{q}$-axis component of the estimated active-flux. This leads to an error signal of nature [9]

$$\epsilon_\theta = \frac{1}{2\, L_\Delta^i\, i_{\hat{d}}} \begin{bmatrix}0\\1\end{bmatrix}^{\mathrm{T}} \Big(\hat{\boldsymbol{\lambda}}_{\hat{d}q} - L_q^i\, \boldsymbol{i}_{\hat{d}q}\Big). \tag{15}$$

Active-flux observer is known to have stability problems are low speeds braking and high speeds motoring regions [9], [10].

2) APP Position Observer: The APP observer is designed within a projection vector framework to maximize the signal-to-noise ratio between the position error signal and the position error [5], [9]. The APP position error signal is given by

$$\epsilon_\theta = \frac{-1}{\omega\, |\hat{\boldsymbol{\lambda}}_{\hat{d}q}^a|^2}\, \big(\hat{\boldsymbol{\lambda}}_{\hat{d}q}^a\big)^{\mathrm{T}}\, \mathbf{J}\, \big(\boldsymbol{G} + \omega\, \mathbf{J}\big)\, \big(\hat{\boldsymbol{\lambda}}_{\hat{d}q} - \boldsymbol{\lambda}_{\hat{d}q}^i\big). \tag{16}$$

where $\hat{\boldsymbol{\lambda}}_{\hat{d}q}^a = \mathbf{J}\, \hat{\boldsymbol{\lambda}}_{\hat{d}q} - \boldsymbol{L}_\partial\, \mathbf{J}\, \boldsymbol{i}_{\hat{d}q}$ is the estimated auxiliary-flux vector.

The APP observer mitigates the stability problems of the active-flux observer. In addition, it is shown to be immune to variations in stator resistance and non-ideal inverter voltage compensation [11].

E. Speed and Position Observer

1) Phase-Locked-Loop: A conventional phase-locked-loop (PLL) with a proportional-integral (PI) controller is employed to drive the position error signal ϵ to zero as

$$\hat{\omega} = k_p\, \epsilon + \int k_i\, \epsilon\, \mathrm{d}t \qquad \hat{\theta} = \int \hat{\omega}\, \mathrm{d}t \tag{17}$$

where k_p and k_i are the respective gains. The gains of the PLL are tuned for a critically damped response considering $\epsilon = \tilde{\theta}$ by placing the poles at $s = -\Omega_\omega$ as $k_p = 2\,\Omega_\omega$ & $k_i = \Omega_\omega^2$.

2) Fusion Mechanism: The position observer is designed to transition from low to high speed model around the cross-over frequency g rad/s, akin to the flux observer. To refrain from sharp discontinuous transition and chattering, the two position estimation models are fused together with a linear speed dependent fusion coefficient f_ω, expressed as

$$\epsilon^k = f_\omega \cdot \epsilon_\theta^k + \big(1 - f_\omega\big) \cdot \epsilon_h^k \tag{18a}$$

$$f_\omega = \begin{cases} 0, & \text{if } |\hat{\omega}^k| < g - \omega_g \\ 1, & \text{if } |\hat{\omega}^k| > g + \omega_g \\ \frac{|\hat{\omega}^k| + \omega_g - g}{2\omega_g}, & \text{otherwise} \end{cases} \tag{18b}$$

where the term ω_g signifies the span of transition on either sides of cross-over frequency g. The fusion coefficient f_ω is designed to ensure smooth transition over the span of speeds $g - \omega_g$ to $g + \omega_g$.

IV. Simulation Results

The switching frequency is set to 10 kHz with synchronous sampling and the simulation step time is 2 μs. The hybrid flux observer gain is $g = 2\pi \cdot 10$ rad/s and the fusion window span is $\omega_g = 2\pi \cdot 4$ rad/s. The speed controller is tuned for critically damped poles at $s = -2\pi \cdot 1$ rad/s. The current controllers are tuned for bandwidth of $\Omega_T = 2\pi \cdot 75$ rad/s.

The sinusoidal voltage injection frequency is $\omega_h = 2\pi \cdot 500$ rad/s and magnitude $V_h = 100$ V. The cut-off frequency of LPF in the signal processing stage for sinusoidal injection in (9a) and (12a) is 50 Hz. The poles of the PLL are tuned for $\Omega_\omega = 2\pi \cdot 10$ rad/s. The frequency of square-wave injection is 5 kHz. A minimum current $i_q^{min} = 0.2$ p.u. is imposed for the saturation of ribs and fundamental excitation at no-load.

The incremental inductance matrix $\boldsymbol{L}_\partial$ is computed in real-time from the flux-map LUTs; as an example:

$$l_d(\boldsymbol{i}_{dq}) = \frac{\boldsymbol{\Lambda}_d(i_d + \delta i_d, i_q) - \boldsymbol{\Lambda}_d(i_d, i_q)}{\delta i_d} \tag{19}$$

where δi_d is a small value (≈ 0.02 p.u.). The other incremental inductances are obtained real-time in a similar fashion.

A. Sensorless Control Validation

Two motors are different sizes are used to demonstrate the generality of this tool. Motor A is a 1.1 kW, 7.3 Nm, 1500 rpm, 3 A and 2 pole pairs. Motor B is 4.4 kW, 17 Nm, 2500 rpm, 15 A and 3 pole pairs. The results in Fig. 6 correspond to the sensorless settings of square-wave voltage injection with current-model flux demodulation at low speeds and APP observer at high speeds region. The fusion coefficient f_ω indicates the relative dominance of the two models. The position error is less than a few electrical degrees in either simulations. It is worth highlighting that the two simulations have similar tuning/calibration, demonstrating the generality and the machine-independent nature of syreDrive.

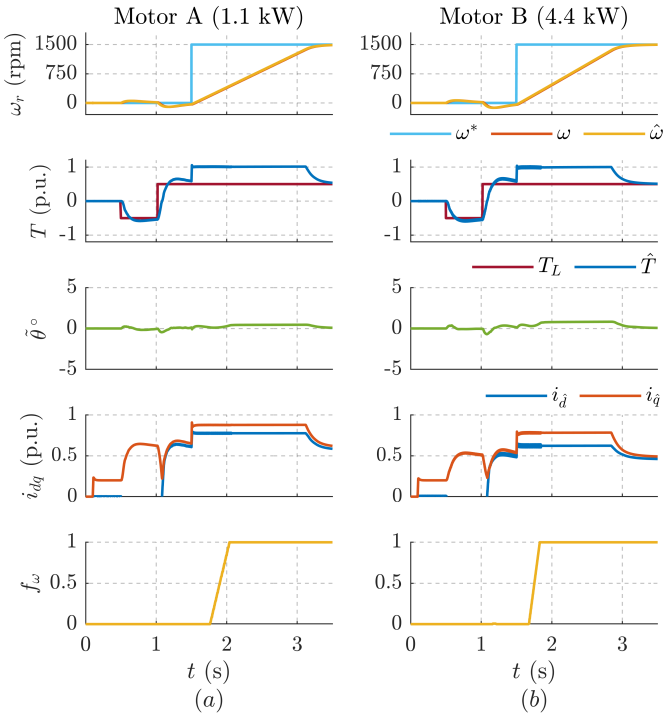

Fig. 6. Simulation result of syreDrive automated sensorless control generation using square-wave voltage injection with current-model flux demodulation at low speeds and APP at high speeds region: (a) Motor A (1.1 kW); (b) Motor B (4.4 kW).

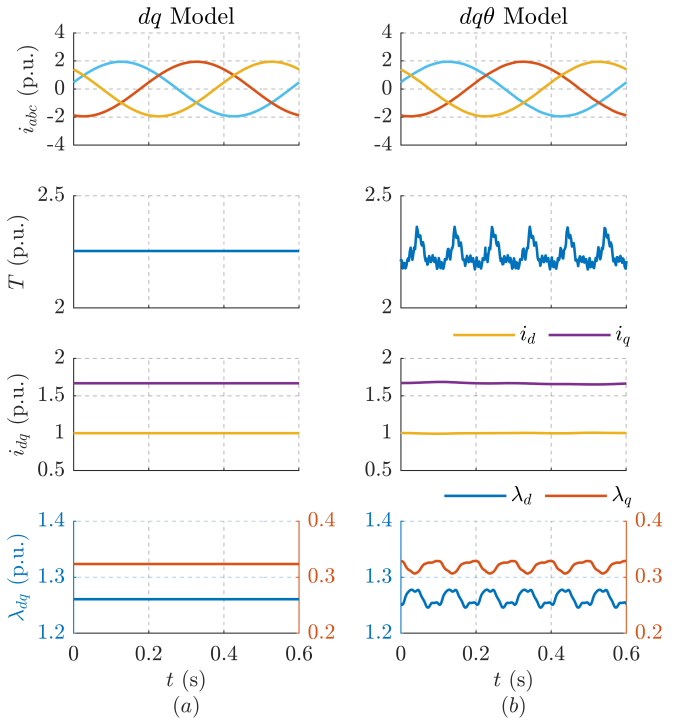

Fig. 7. Simulation result using encoder at current control and constant speed $\omega_r = 50$ rpm: (a) 2D dq flux maps; (b) 3D $dq\theta$ flux maps.

B. Significance of $dq\theta$ Flux Maps

The simulation result in Fig. 7 shows the current control at constant speed using encoder to illustrate the comparison between the averaged 2D dq flux maps and the position-dependent 3D $dq\theta$ flux maps. The torque is computed using the average model (3) in Fig. 7(a) while it is computed from the FEA-based 3D $dq\theta$ torque map in Fig. 7(b).

In Fig. 7(b), at the low speed of $\omega_r = 50$ rpm, the dominant space harmonics are within the bandwidth of the current controller. Thus, a constant dq stator currents are established while the harmonics are reflected in the stator fluxes in Fig. 7(b). In particular, the 6^{th} harmonic in the torque ripple is evident. It is reiterated that the computation cost of using the 3D $dq\theta$ model is only about 10% higher than the 2D dq model due to the efficient method of interpolation.

V. Conclusion

This paper discussed a new feature, syreDrive, for automated sensorless control Simulink model generation within the motor design framework of SyR-e. A general template of sensorless control code is designed with machine-independent turning guidelines that is automatically customized to the motor prototype under inspection. This allows for quick evaluation of sensorless control capability of prototypes through a simple GUI, following the machine design tool-chain. Thus, it serves as a valuable open-source tool for engineers in design and development of sensorless-friendly motor drive systems.

References

[1] F. Cupertino and G. Pellegrino, "SyR-e: Synchronous Reluctance (machines) - evolution." [Online]. Available: www.github.com/SyR-e

[2] D. Meeker, "FEMM: Finite Element Method Magnetics." [Online]. Available: www.femm.info

[3] "Mathworks." [Online]. Available: www.mathworks.com

[4] Y. D. Yoon, S. K. Sul, S. Morimoto, and K. Ide, "High-bandwidth sensorless algorithm for AC machines based on square-wave-type voltage injection," *IEEE Transactions on Industry Applications*, vol. 47, no. 3, pp. 1361–1370, 2011.

[5] A. Varatharajan, P. Pescetto, and G. Pellegrino, "Sensorless Synchronous Reluctance Motor Drives: A Full-Speed Scheme using Finite-Control-Set MPC in a Projection Vector Framework," *IEEE Transactions on Industry Applications*, vol. 56, no. 4, pp. 3809–3818, 2020.

[6] T. Tuovinen and M. Hinkkanen, "Adaptive Full-Order Observer With High-Frequency Signal Injection for Synchronous Reluctance Motor Drives," *IEEE Journal of Emerging and Selected Topics in Power Electronics*, vol. 2, no. 2, pp. 181–189, 2014.

[7] E. Capecchi, P. Guglielmi, M. Pastorelli, and A. Vagati, "Position-sensorless control of the transverse-laminated synchronous reluctance motor," *IEEE Transactions on Industry Applications*, vol. 37, no. 6, pp. 1768–1776, 2001.

[8] I. Boldea, M. C. Paicu, G. Andreescu, and F. Blaabjerg, ""Active Flux" DTFC-SVM Sensorless Control of IPMSM," *IEEE Transactions on Energy Conversion*, vol. 24, no. 2, pp. 314–322, 2009.

[9] A. Varatharajan and G. Pellegrino, "Sensorless Synchronous Reluctance Motor Drives: A General Adaptive Projection Vector Approach for Position Estimation," *IEEE Transactions on Industry Applications*, vol. 56, no. 2, pp. 1495–1504, 2020.

[10] M. Hinkkanen, S. E. Saarakkala, H. A. A. Awan, E. Mölsä, and T. Tuovinen, "Observers for Sensorless Synchronous Motor Drives: Framework for Design and Analysis," *IEEE Transactions on Industry Applications*, vol. 54, no. 6, pp. 6090–6100, 2018.

[11] A. Varatharajan and G. Pellegrino, "Sensorless synchronous reluctance motor drives: A projection vector approach for stator resistance immunity and parameter adaptation," *IEEE Transactions on Industry Applications*, vol. 56, no. 5, pp. 5003–5012, 2020.