The dq-theta Flux Map Model of Synchronous Machines

The dq-theta Flux Map Model of Synchronous Machines

Simone Ferrari, Gaetano Dilevrano, Paolo Ragazzo and Gianmario Pellegrino
Energy Department "Galileo Ferraris"
Politecnico di Torino
Turin, Italy
simone.ferrari@polito.it, gaetano.dilevrano@polito.it, paolo.ragazzo@polito.it, gianmario.pellegrino@polito.it

***Abstract*—This paper presents the $dq\theta$ flux and torque maps modelling approach for synchronous machines, where the dependency of flux linkage and torque respect to the rotor phase angle θ is recorded to account for the effects of space harmonics and torque ripple. Using a FEA or experimental flux map identification session over a grid of i_d, i_q operating points, the new $dq\theta$ maps are obtained via dedicated manipulation of the test results. Instead of averaging the dq flux linkage and torque waveform respect to θ, the θ related information, which is flux linkage undulation and torque ripple, is retained. The fundamental - or θ averaged - dq flux maps can be thus considered a subset of the extended $dq\theta$ approach. Magneto-static FEA simulations are used in the paper for describing the storage and manipulation procedure. With the new maps, the torque and flux waveforms are retrieved by bi-linear interpolation of the respective maps at any i_d, i_q condition, including non-identified operating points. Moreover, the torque and flux maps of the machine after skewing can be calculated off-line in seamless time by off-line interpolation of the $dq\theta$ maps, without additional FEA simulation and for any skewing angle and number of slices. As a further validation, the presented procedures will be included into an open-source design environment, available on the web.**

I. Introduction

Today, the use of permanent magnet (PM) synchronous machines is increasing at high pace in many applications, especially in traction [1]. This includes interior PM and surface-mounted PM machine types [2], [3], but also synchronous reluctance (SyR) and PM-assisted SyR machines [4]. The flux linkage maps $\lambda_d(i_d, i_q)$, $\lambda_q(i_d, i_q)$ are a general way of modelling all such machines [5], used for computation of the steady-state and dynamic performance. The flux maps can be evaluated via Finite Element Analysis (FEA) or experiments. Notably, the literature refers to the fundamental dq flux maps, which contain solely the average values of the flux linkages with respect to the rotor phase angle, omitting their undulation and therefore the effect of space harmonics. Although the θ information is usually present in the FEA simulated output used for obtaining the fundamental model, or in the experimental raw data, provided that the data recorder in use memorizes also the rotor position.

In [6] the harmonic content is included in the FEA-evaluated flux maps by using the Fourier series and polynomial functions. In [7], the flux linkage harmonics are segregated using the frozen permeability method. Besides being computationally intensive and to some extent approximated, both the mentioned works focus on improving the torque ripple computation and do not mention all the possible advantages of the extended machine parametrization.

This work formalizes the data processing and organization needed to include the effect of the rotor phase angle in the machine flux and torque maps, using the same set of magneto-static FEA data previously used for populating the dq fundamental flux maps. A SyR machine for industry applications will be used as example. Fig. 1 reports the cross-section of the considered motor, whose ratings are reported in Table I. The evaluation of the torque waveform in a non-identified operating point and the off-line computation of the machine figures after skewing are presented as highlights of the potential associated to the $dq\theta$ approach. All the presented procedures are incorporated and openly accessible in the Matlab-based design software SyR-e [8].

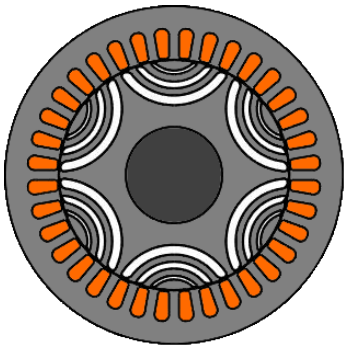


Fig. 1: Cross-section of the SyR machine selected as example.

TABLE I: SyR motor ratings

Nominal current	i_{nom}	15	[Apk]
Max current	i_{max}	30	[Apk]
Nominal torque	T_{nom}	17	[Nm]
Max torque	T_{max}	43	[Nm]
DC link voltage	V_{dc}	565	[V]
Nominal speed	n_{nom}	2500	[rpm]
Max speed	n_{max}	6000	[rpm]
Nominal power	P_{nom}	4.4	[kW]
Max power	P_{max}	11.2	[kW]
Number of pole pairs	p	3	

II. FEA Flux Maps Computation and Organization

Flux maps are a very general way to represent the magnetic model of a synchronous motor. Usually, the fundamental dq flux linkages are expressed as a function of the dq currents of the motor. The average electromagnetic torque can be retrieved by cross-product of the flux linkage and current components, or by a dedicated map, function of the same current coordinated. The flux maps can be computed through FEA simulation [9], [10] or measured with dedicated experimental procedures [11], [12]. In the following, the guidelines for fast FEA computation of flux maps will be reviewed, looking at the tradeoff between accuracy and computational time. Then, the fundamental dq flux maps will be presented as a prologue to the new extended $dq\theta$ model.

A. Fast FEA Evaluation of dq flux and torque

Reference is made to the 2D magnetostatic FEA simulation of radial-flux synchronous machines. The same approach applies to 3D and transient FEA, as well as to synchronous machines of different types.

Geometric and electric symmetries must be utilized, for fast simulation. Referring to the motor in use, the anti-periodic symmetry applies, allowing the simulation of one pole, which is $1/6$ of the complete motor and a rotation of 60 electrical degrees, which is again $1/6$ of the electrical period.

Under these assumptions, the simulation of one i_d, i_q operating point with a mesh of 5095 elements, using 30 positions over 60 electrical degrees of rotation takes about 75 seconds. Fig. 2 shows the flux linkage and torque waveform output of the simulation, giving evidence of the 30 simulated points (red). The full-cycle waveform (continuous blue lines) are obtained by symmetry.

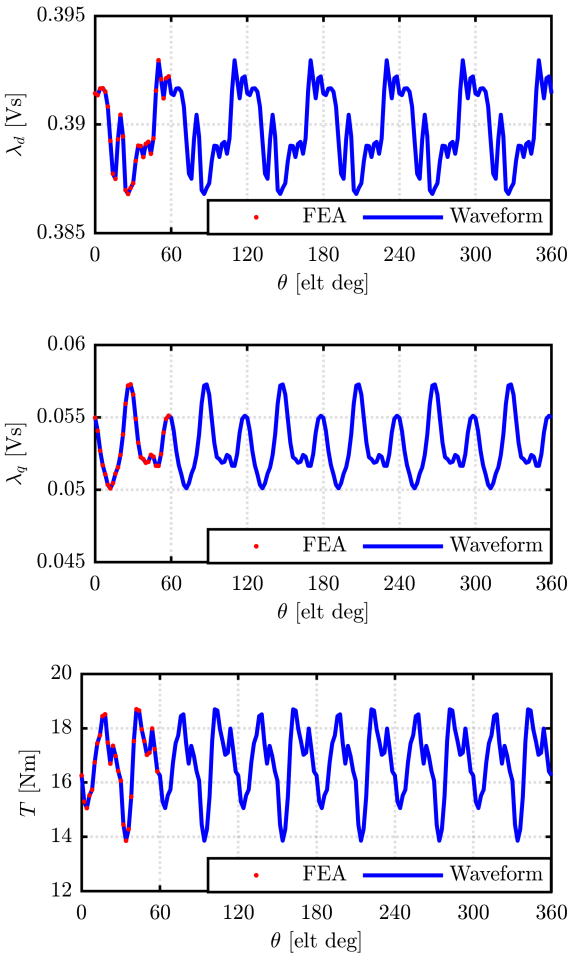


Fig. 2: Waveform for d and q axis flux linkages and torque: FEA point simulated in red and complete waveform in blue.

B. Fundamental dq Flux Maps Organization

The fundamental flux maps are computed by repeating the 60-degrees simulation described above on a regular grid in the (i_d, i_q) domain. The numbers of considered i_d and i_q values are called n_d and n_q, respectively. The resulting flux linkages λ_d, λ_q and torque T are averaged with respect to θ and saved in corresponding look-up tables.

The flux map functions are denoted with capital bold symbols, so $\mathbf{\Lambda}_d(i_d, i_q)$ and $\mathbf{\Lambda}_q(i_d, i_q)$ are the average dq flux linkages maps, while $\mathbf{T}(i_d, i_q)$ is the average torque map. Further matrices can be computed, for example the peak-to-peak torque ripple map $\mathbf{\Delta T}(i_d, i_q)$.

The 2-dimensional look-up tables (LUTs) are organized as follows. The grid of current values is organized as (1):

$$\mathbf{I}_d = \begin{bmatrix} i_{d,1} & \dots & i_{d,n_d} \\ \dots & \dots & \dots \\ i_{d,1} & \dots & i_{d,n_d} \end{bmatrix}, \mathbf{I}_q = \begin{bmatrix} i_{q,1} & \dots & i_{q,1} \\ \dots & \dots & \dots \\ i_{q,n_q} & \dots & i_{q,n_q} \end{bmatrix} \quad (1)$$

The λ_d look-up table is defined accordingly (2).

$$\mathbf{\Lambda}_d = \begin{bmatrix} \lambda_d(i_{d,1}, i_{q,1}) & \dots & \lambda_d(i_{d,n_d}, i_{q,1}) \\ \dots & \dots & \dots \\ \lambda_d(i_{d,1}, i_{q,n_q}) & \dots & \lambda_d(i_{d,n_d}, i_{q,n_q}) \end{bmatrix} \quad (2)$$

The same operator $\mathbf{\Lambda}_d$ is used to indicate both the look-up table and the flux map function $\mathbf{\Lambda}_d(i_d, i_q)$ for the sake of simplicity. The values of the function are obtained from the look-up table by bi-linear interpolation. The other maps are not explicitly reported for brevity.

The dq current grid must cover the overload condition. Reasonable numbers for simulated n_d and n_q are between 10 and 20, i.e. tables of 100 to 400 elements. Fig. 3 reports an example of flux and torque maps, where the simulated points are highlighted by red dots. $n_d = n_q = 15$ was used in this case.

The number of simulated rotor positions n_θ can be as low as 6 for fundamental model flux maps. However, for a precise evaluation of the pk-pk torque ripple map it is recommended to use a value of n_θ as high as 30 (on 60 electrical degrees). The same recommendation is valid for the $dq\theta$ maps described later.

Altogether, the number of FEA instances $n_d \cdot n_q \cdot n_\theta$ determines the total computational time. In the reported example, $n_d \cdot n_q = 15 \cdot 15 = 255$ current points and $n_\theta = 30$ rotor positions are used, for a total of 6750 FEA simulations.

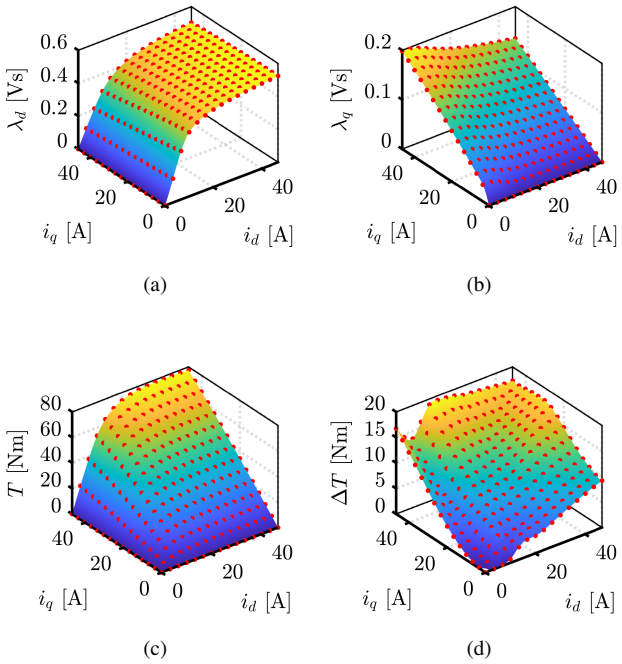


Fig. 3: FEA maps of the motor under test. a) d-axis flux linkage, b) q-axis flux linkage, c) average torque, d) peak-to-peak torque ripple.

Parallel computing can further speed up the process, as different dq operating points can be evaluated in parallel by different cores. Using the 6 cores of a laptop with an Intel i7-8750H CPU and 16 GB of RAM the 6750 simulations take 55 minutes circa. This set of data is used for the dq as well as for the $dq\theta$ maps.

Finally, the dq flux maps are re-sampled over a finer mesh of $n_d = n_q = 256$ current levels using Matlab interpolation. The maps on the finer mesh are reported in color in Fig. 3.

C. Novel $dq\theta$ Flux Maps Organization

Described as above, the fundamental dq flux maps are a lossy way to store the results of the simulations, as the rotor position information is lost when averaging the flux linkage and torque waveforms. The $dq\theta$ approach can be seen as a better organization of the same simulated data, to avoid information loss. The data are stored in 3-dimensional look-up tables, to have the dependency from d- and q-axis currents and the rotor phase angle. The $dq\theta$ domain is defined with reference to the d- and q-axis current steps Δi_d, Δi_q and the rotor angle step $\Delta\theta$, and using i, j and k as indexes along the respective dimensions of the look-up table:

$$\begin{cases} i_d(i,j,k) = (i-1)\cdot\Delta i_d \\ i_q(i,j,k) = (j-1)\cdot\Delta i_q \\ \theta(i,j,k) = (k-1)\cdot\Delta\theta \end{cases} \tag{3}$$

The example refers to the grid of currents starting from zero, valid for the reported case of a SyR machine. For PM synchronous machines the current domain is suitably redefined. For example, the d flux linkage is reported in Fig. 4 as a function of the three input variables using colored maps at different values of θ. The red dots are the FEA points. The current and angle steps of simulation are the same used for the fundamental maps, according to the declared values of n_d, n_q, n_θ.

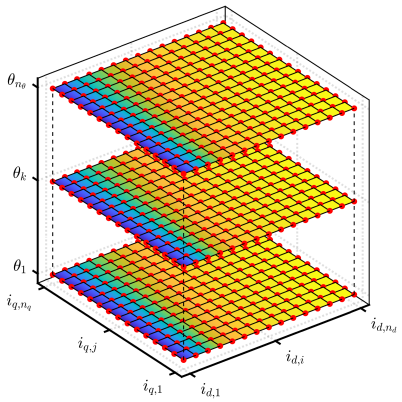


Fig. 4: Composition of the 3D look-up table of the d-axis flux linkages, by stacking 2D tables. Three values of rotor position are shown for clarity.

D. $dq\theta$ Maps Interpolation

As for the fundamental flux maps, the same symbols are used both for functions and for the corresponding look-up tables. For instance $\mathbf{\Lambda}_d(i_d, i_q, \theta)$ identifies the d-axis flux linkage function, and its values are obtained by interpolating the $\mathbf{\Lambda}_d$ look-up table.

To ease the data manipulation, the θ domain is extended from the simulated 60 electrical degrees to one electrical period (360 degrees), by exploiting the waveforms symmetry discussed in section II-A. Hence, assuming n_d and n_q current levels, respectively in the d and q axis, a n_θ simulated rotor positions provide $dq\theta$ look-up tables with the dimension of $n_d \times n_q \times 6n_\theta$.

Instead of interpolating on a super-fine mesh as done for the fundamental maps, the $dq\theta$ maps are accessed and manipulated using the `griddedInterpolant` function of Matlab [13]. This function creates interpolant objects $\mathbf{\Lambda}_d(i_d, i_q, \theta)$, $\mathbf{\Lambda}_q(i_d, i_q, \theta)$ and $\mathbf{T}(i_d, i_q, \theta)$ from the set of simulated data. In this way, any (i_d, i_q, θ) point within the identification domain can be evaluated at high resolution, provided that the identification mesh is well conceived (appropriate number of simulated points).

E. Computation of the Inverse Flux Maps

The inverse of the flux maps, i.e. the maps of currents function of flux linkages $\mathbf{I}_d(\lambda_d, \lambda_q)$, $\mathbf{I}_q(\lambda_d, \lambda_q)$ (or $\mathbf{I}_d(\lambda_d, \lambda_q, \theta)$, $\mathbf{I}_q(\lambda_d, \lambda_q, \theta)$), are evaluated by manipulation of the dq (or $dq\theta$) maps. The inverse maps are used to solve the dynamic model of the machine when the flux linkages are the state variables,

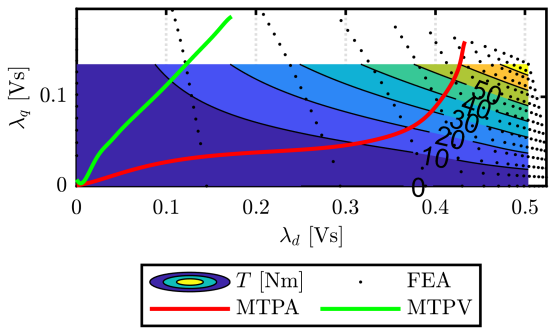


Fig. 5: Inverse torque map (colored contour), FEA-simulated points reported in black, MTPA and MTPV locus reported in red and green, respectively.

for example for control simulation purposes, as shown later. Given the rectangular (i_d, i_q) mesh, the corresponding (λ_d, λ_q) points cover the distorted domain shown in Fig. 5, where the black dots represent the FEA-simulated points (i.e. the regular grid of current points). The input domain of the inverse map LUTs is thus reduced to a regular mesh of (λ_d, λ_q) points, and this dictates loss of information at the borders. The figure shows in color the domain of the inverse LUTs, and in white the non represented areas. The interpolation of the raw data to the regular (λ_d, λ_q) mesh is done using the `scatteredInterpolant` function of Matlab [14].

Fig. 6 shows the current and torque maps function of the flux linkage components of the benchmark motor.

The flux maps inversion is extended to the $dq\theta$ model by applying the map inversion to each rotor position. The surfaces of the inverse maps for a single angular value are similar to the ones reported in Fig. 6 for the fundamental model.

III. Review of Fundamental Model Map

Examples of fundamental dq maps manipulation are briefly reviewed in this section:

A) computation of the control trajectories such as MTPA (Maximum Torque per Ampere), MTPV (Maximum Torque per Voltage) and related operating limits (torque versus speed profile at limited current and voltage amplitudes)

B) motor model scaling:
- number of turns: the flux and torque maps of the rewound machine can be derived without the use of FEA, by scaling the current according to the inverse turns ratio and the flux linkage according to the turns ratio.
- stack-length: the flux and torque maps of the shorter or longer machine are derived by scaling the flux linkages according to the stack-length ratio.

All such features are still valid also for the $dq\theta$ maps and they are part of the dedicated SyR-e interface called

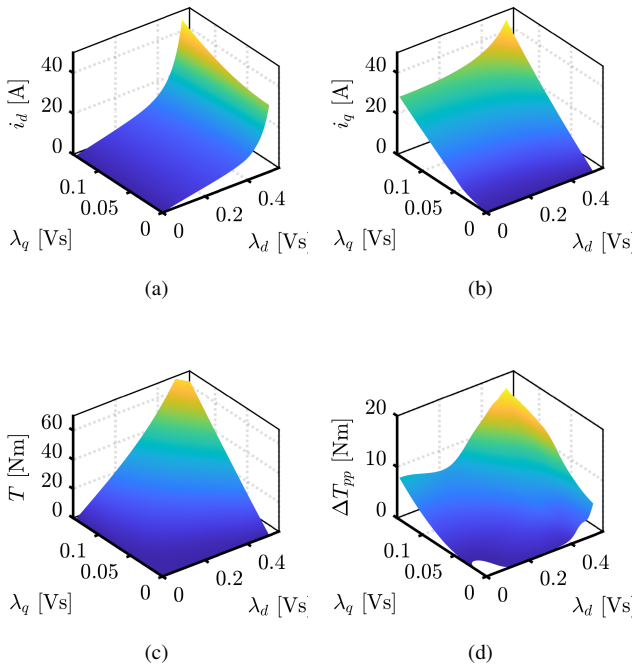


Fig. 6: Inverse flux maps: d-axis (a) and q-axis (b) currents, torque (c) and torque ripple (d).

GUI_Syre_MMM (Graphical User Interface of SyR-e for the Magnetic Model Manipulation).

A. Identification of the Control Trajectories

One manipulation of primary importance is the control trajectory identification. This is done using the dq flux maps, referring to average values as typical for control. The most considered trajectories are the MTPA and the MTPV laws, but other loci can be identified, as the maximum power factor per ampere or the minimum torque ripple law. The method to identify one locus is general and it is usually defined as the maximization or minimization of the ratio of two figures across the i_d, i_q domain.

For example, the MTPA consists of the maximization of the average torque for a given current amplitude. The points of the map belonging to the selected current contour are isolated and the torque is evaluated from the torque maps. The maximum torque (i_d, i_q) combination is found and stored as one MTPA element. The process is repeated for the next current amplitude, ranging from zero to the maximum allowed by the maps current domain. The MTPV is computed using the same algorithm but moving on constant flux linkage curves instead of the current ones.

The MTPA and MTPV trajectories of the SyR machine are reported in Fig. 7. They are expressed as i_d and i_q vectors function of torque T. This allows to extract from the flux and torque maps all the other information, as flux linkages (dq components, amplitude, angle), torque ripple and so on.

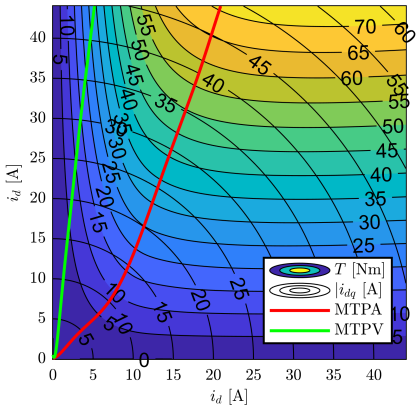


Fig. 7: MTPA (red) and MTPV (blue) locus on the torque map, computed with the algorithm included in SyR-e.

B. Flux Maps Scaling

The electric motor scaling can be a useful tool in the design section, that avoid the FEA simulation of a scaled machine. Several scaling rules are available [15], at this time the manipulations embedded in SyR-e are:

- change the number of turns;
- change the axial length;
- add extra inductance.

Axial length and number of turns are independent from the FEA simulation, since the model is 2D (axial length is a multiplication factor) and the simulation is set with one turn each phase. Dealing with the additional inductance, it can be added to the flux linkages matrices $\mathbf{\Lambda}_d$ and $\mathbf{\Lambda}_q$, both for dq and $dq\theta$ models. Torque matrix is not affected from additional inductance since it is equal on both axis.

IV. New $dq\theta$ maps manipulation

The additional possibilities enabled by the $dq\theta$ model are:

A) computation of the flux linkages and torque waveforms of non-simulated i_{dq} conditions.
B) computation of the step-skewed motor model, for any number of steps and any angle of skew.
C) dynamic model of the machine accounting for back-emf and torque harmonics.

A. Torque and Flux Waveforms

One of the main advantage of the $dq\theta$ model is the possibility to retrieve the flux linkage and torque waveform without running further FEA simulations. The waveform are obtained by interpolation from the respective 3D matrix of the $dq\theta$ model, with inputs constant i_d and i_q vectors and θ vector spanning on the whole electrical period.

Fig. 8 compares the dq flux linkages and torque waveform computed with FEA and interpolated from $dq\theta$ model. The selected point (i_d, i_q) is close to the MTPA at rated current, but it is far from the FEA-evaluated point of the flux map- in order to test the interpolation capability. The waveform from the two methods are perfectly overlapped, even if the selected operating point is not a node of the identification grid. However, the match between the two methods can be reduced if the set of FEA simulations used to create the $dq\theta$ model is limited. The presented results achieve a good level of accuracy with the maps of $n_d \times n_q \times n_\theta = 15 \times 15 \times 30$, computed in about 55 minutes of FEA simulations on a standard laptop, and represents a good computational time compared to the accuracy.

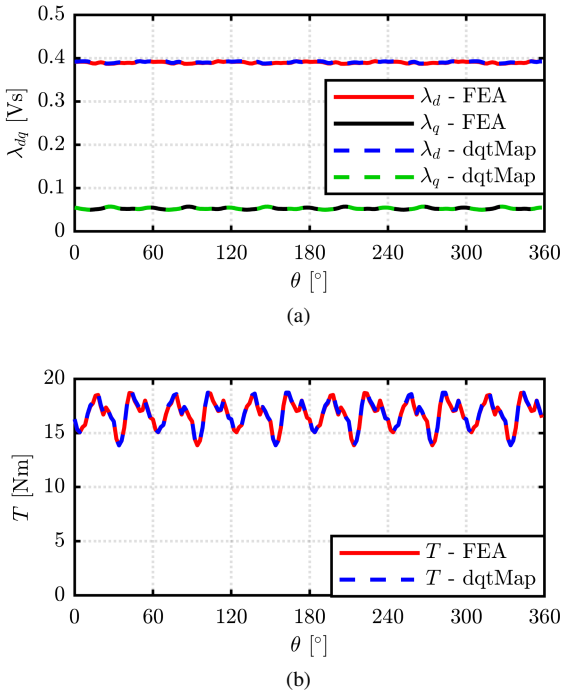


Fig. 8: Validation of the $dq\theta$ waveform interpolation: flux linkages (a) and torque (b) waveform FEA evaluated (dashed lines) and interpolated from $dq\theta$ flux maps (solid lines).

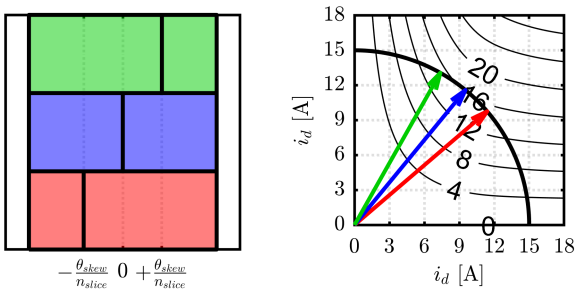
B. Evaluation of the Skewed Motor Performance

Skewing is one of the most common method to reduce torque ripple for electric machines. It consists of an axial rotation of the rotor laminations of a well defined angle, called skew angle θ_{skew}. Assuming a target torque ripple harmonic h, the mechanical skew angle in degrees is:

$$\theta_{skew} = \frac{360}{p \cdot h} \tag{4}$$

For PM motors, the most common type of skewing is the step-skewing, that consist in dividing the rotor in n_{slice} axial slices and rotating one from the other, in order to emulate a continuous skewing, that can be seen as an extreme case, with the number of slices equal to the number of laminations.

Ideally, to evaluate the performance of a skewed motor, the 3D FEA simulation is needed because of the 3D nature of the problem [16]. Moreover, a simpler method, called multi-slice simulation, can be implemented [17]. Assuming n_{slice} axial slices, the method run n_{slice} 2D FEA simulations (one



(a) (b)

Fig. 9: Example of a $n_{slice} = 3$ step-skewing: axial sketch of the motor (a) and currents on the dq plane for each slice (b).

per axial section) and then sum the simulations results. The initial position of each slice is shifted from the other, and the current is the same for each simulation. This means that the dq currents that must be set in each slice are slightly different, because of the angular shift. The concept is described in the following for a $n_{slice} = 3$ step-skewed motor.

Fig. 9a reports a simplified top view of the motor, with the three slices highlighted with different colors and a black reference line to highlight the skewing. Since n_{slice} is odd, the dq reference of the central slice is the same of the full machine, while the dq axis of the other slices are slightly rotated because of the skewing. The rotation of each slice respect to the previous is $\frac{\theta_{skew}}{n_{slice}}$. Fig. 9b shows the same dq current on the local dq axis of each slice, and the colors coherent with Fig. 9a. The effect of the angular skewing is the rotation of the dq currents too. The blue section is aligned with the whole machine dq axis, so has no shift.

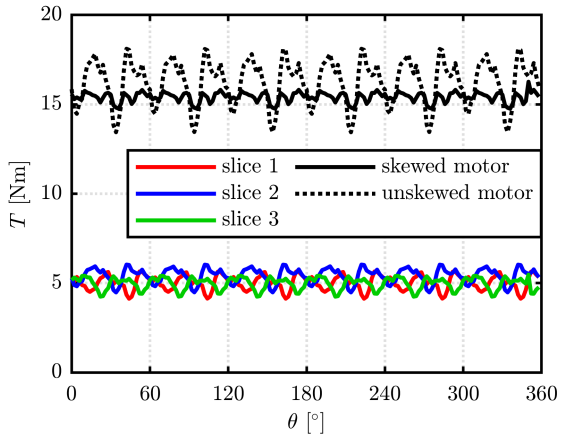


Fig. 10: Torque waveform of the step-skewed motor: single slice waveform (colored), skewed motor waveform (solid black) and unskewed motor waveform (dotted black).

Fig. 10 shows the torque contribution of the three slices and the torque waveform of the step-skewed motor. The difference between the three colored waveform are:

- an angular shift of the peaks, according to θ_{skew};
- slightly different torque profile, caused from the different local dq working point.

The waveform of the skewed motor is obtained as the sum of the colored waveform and it is reported with black solid line. For a sake of comparison, the unskewed motor waveform is reported with black dotted line, to highlight the torque ripple reduction and the average torque detriment caused from the skewing.

The flux maps evaluation of a skewed motor with FEA simulations is n_{slice} time longer that the flux maps evaluation of a straight motor, for the reason introduced before. Moreover, the possibility to retrieve torque and flux linkage waveform from the proposed $dq\theta$ flux maps can drastically speed-up the evaluation process.

With $dq\theta$ model, skewed motor flux maps are computed in post-processing from the straight motor flux maps. The computational time for the post-processing is around one minutes, that is negligible compared with the time needed to compute the flux map of the straight motor (around 45 minutes). Thanks to the $dq\theta$ post-processing, it is possible to evaluate the flux maps of skewed motor in minutes instead of hours of simulation.

The $dq\theta$ post-processing is equivalent to the multi-slice simulation, with the difference that no FEA simulations are run and all the information are extracted from $dq\theta$ flux maps, by selecting proper dq currents and proper rotor angle. For each slice, the respective flux maps are computed by accounting for the shorter axial length. In the interpolation process, attention must be paid on the dq reference system when summing the contributions from the different slices. The dq axis rotation between the local and global reference system cause a reduction of the dq current domain of the skewed flux maps. Fig. 11 shows this reduction for $\theta_{skew} = 10°$ and $n_{slice} = 3$ applied to the benchmark motor: the domain of the unskewed motor (equal to the central slice) is highlighted in blue, while the domain of the two rotated slices are reported with black lines. The dq domain of the skewed motor flux maps (colored in red) are identified as the biggest square included in all the slice domains.

C. Results on Skewed Motor

After the skewing post-processing, the flux maps (both dq and $dq\theta$) have the same format of the baseline motor, and thus all the elaboration described in the previous sections can be applied also to skewed motor model. The main advantage of the $dq\theta$ model is the availability of the torque waveform for both unskewed and skewed motor, without the need of further simulations. An example is reported in Fig. 12, where the torque waveform computed from $dq\theta$ maps of unskewed and skewed motor are compared, for two different current levels. These results are istantaneously obtained directly from the $dq\theta$ models of the baseline and skewed motor.

Since the format of the skewed motor flux maps is the same of the baseline motor, all the post-processing operations can be applied also to skewed motor. For instance, Fig. 13 reports

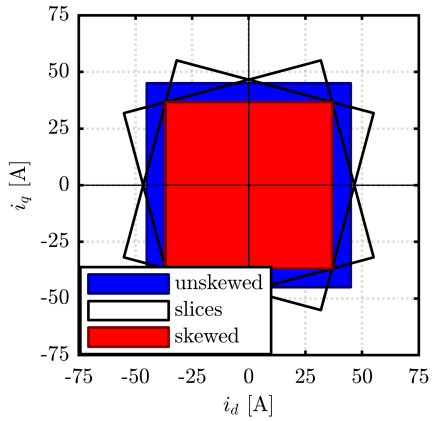


Fig. 11: Limits of the flux maps of skewed motor computed with post-processing: unskewed motor flux map in blue and skewed motor flux map in red.

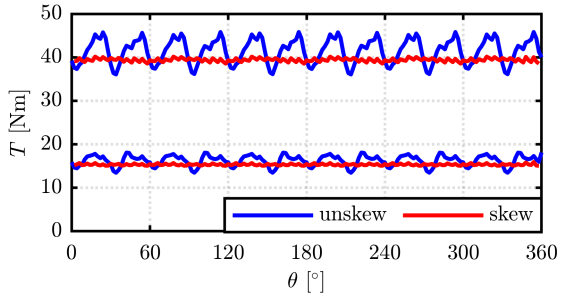


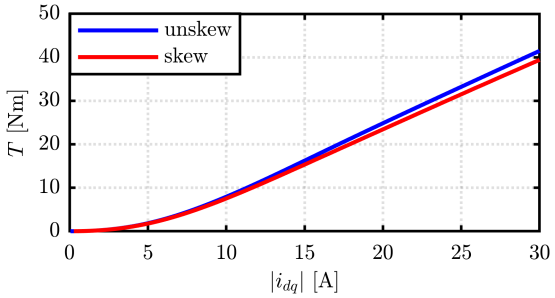
Fig. 12: Torque waveform for unskewed (blue) and skewed (red) motors, retrieved from $dq\theta$ model for $|I_{dq}| = 15\,\text{A}$ and $|i_{dq}| = 30\,\text{A}$.

the $T = f(|i_{dq}|)$ and the $\Delta T_{pp} = f(T)$ characteristic of the unskewed and skewed motors, computed along the MTPA (slightly different for the two machines). As expected, the skewed motor expresses slightly lower torque but it achieves a drastic torque ripple reduction.

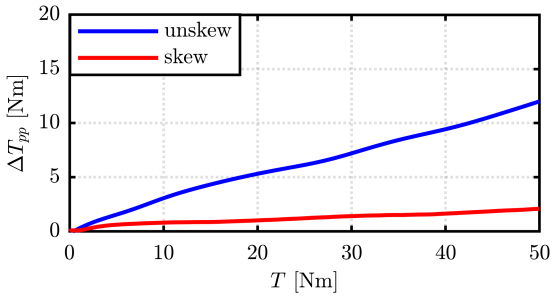
D. Dynamic Model

The $dq\theta$ flux maps can be exploited to build the dynamic model of the machine for control simulation for example in Matlab/Simulink [18]. The dynamic model uses the inverse flux maps, computed as in section II-E, implemented as lookup tables (LUTs). The reference block diagram is in Fig. 14. The use of the inverse $dq\theta$ maps includes the dependency on the rotor position, with a negligible increase of computational time.

A meaningful example is furnished in Fig 15-16. Current vector control in synchronous coordinates is imposed with a reference amplitude of 15 A, first using the dq fundamental model, Fig 15, then using the $dq\theta$ model, Fig. 16. The sinusoidal three-phase currents of Fig 15 are considered in both cases, though, the dq model provides just the fundamental values of flux linkage and thus continuous torque, whereas



(a)



(b)

Fig. 13: Torque versus current (a) and torque ripple versus torque (b) comparison for unskwewed (blue) and skewed (red) motor, along the respective MTPA trajectories.

the $dq\theta$ model returns the more realistic undulated waveforms. Here, the worth of the novel method is highlighted, since contemplating the current and torque undulations is critical in the control domain.

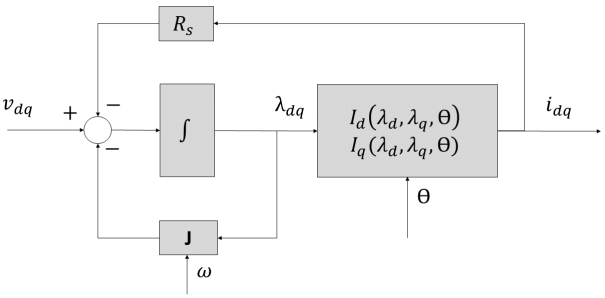


Fig. 14: Dynamic model of the machine, using the $dq\theta$ model.

V. Conclusion

The paper reviews the flux maps evaluation and elaboration of synchronous machines and proposes the improved $dq\theta$ tables model. With the same FEA computational effort, information on harmonic content is retained in the tables, so to memorize torque and flux linkage waveforms at every operating point. Examples of applications include the fast computation of skewed motors, with a time saving of hours in the evaluation of flux maps of a single skewed motor and the $dq\theta$ dynamic model of the machine used for control simulation.

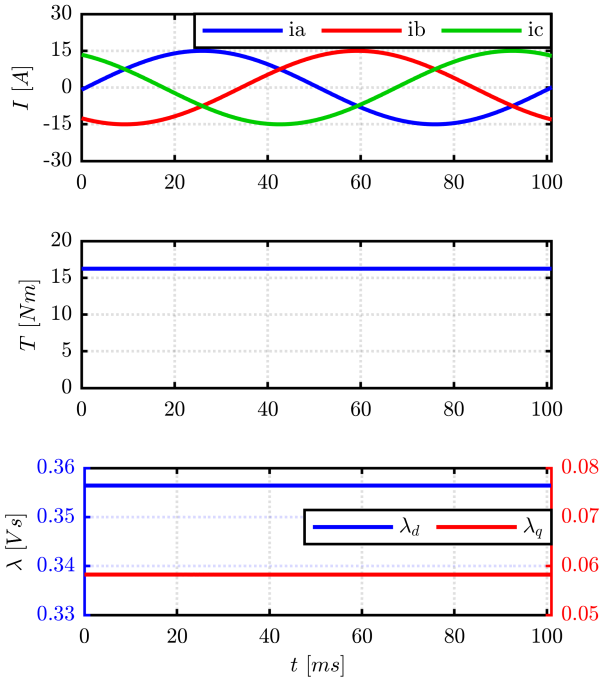


Fig. 15: Field oriented control with fundamental model of the machine.

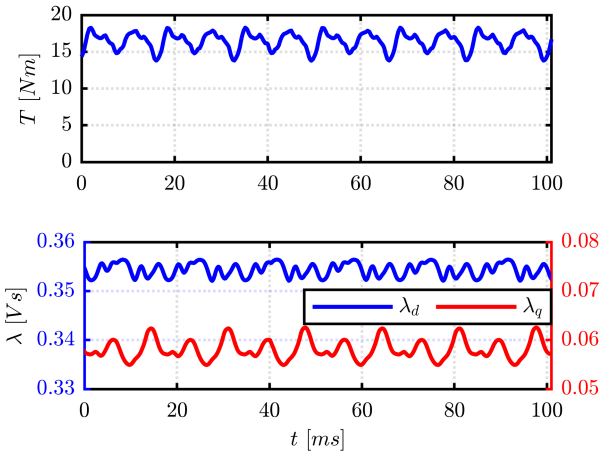


Fig. 16: Field oriented control with $dq\theta$ model of the machine.

The proposed $dq\theta$ approach and all the presented manipulation methods are included in the open-source framework SyR-e, available online.

Acknowledgment

The research has been conducted with the support of Power Electronics Innovation Center (PEIC) of Politecnico di Torino.

References

[1] A. Krings and C. Monissen, "Review and Trends in Electric Traction Motors for Battery Electric and Hybrid Vehicles," in *2020 International Conference on Electrical Machines (ICEM)*. Gothenburg, Sweden: IEEE, Aug. 2020, pp. 1807–1813. [Online]. Available: https://ieeexplore.ieee.org/document/9270946/

[2] D. G. Dorrell, A. M. Knight, M. Popescu, L. Evans, and D. A. Staton, "Comparison of different motor design drives for hybrid electric vehicles," in *2010 IEEE Energy Conversion Congress and Exposition*, Sep. 2010, pp. 3352–3359, iSSN: 2329-3748.

[3] G. Pellegrino, A. Vagati, P. Guglielmi, and B. Boazzo, "Performance Comparison Between Surface-Mounted and Interior PM Motor Drives for Electric Vehicle Application," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 2, pp. 803–811, Feb. 2012.

[4] N. Bianchi, S. Bolognani, E. Carraro, M. Castiello, and E. Fornasiero, "Electric Vehicle Traction Based on Synchronous Reluctance Motors," *IEEE Transactions on Industry Applications*, vol. 52, no. 6, pp. 4762–4769, Nov. 2016.

[5] Eric Armando, R. I. Bojoi, P. Guglielmi, G. Pellegrino, and M. Pastorelli, "Experimental Identification of the Magnetic Model of Synchronous Machines," *IEEE Transactions on Industry Applications*, vol. 49, no. 5, pp. 2116–2125, Sep. 2013. [Online]. Available: http://ieeexplore.ieee.org/document/6504754/

[6] I. Jeong and K. Nam, "Analytic Expressions of Torque and Inductances via Polynomial Approximations of Flux Linkages," *IEEE Transactions on Magnetics*, vol. 51, no. 7, pp. 1–9, Jul. 2015.

[7] A. J. Piña, P. Pramod, R. Islam, R. Mitra, and L. Xu, "Extended model of interior permanent magnet synchronous motors to include harmonics in d-and q- axes flux linkages," in *2015 IEEE Energy Conversion Congress and Exposition (ECCE)*, Sep. 2015, pp. 1864–1871, iSSN: 2329-3748.

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

[9] K. Drobnic, L. Gašparin, and R. Fišer, "Fast and Accurate Model of Interior Permanent-Magnet Machine for Dynamic Characterization," *Energies*, p. 20, 2019.

[10] N. Bianchi and S. Bolognani, "Magnetic models of saturated interior permanent magnet motors based on finite element analysis," in *Conference Record of 1998 IEEE Industry Applications Conference. Thirty-Third IAS Annual Meeting (Cat. No.98CH36242)*, vol. 1, Oct. 1998, pp. 27–34 vol.1, iSSN: 0197-2618.

[11] G. Pellegrino, T. M. Jahns, N. Bianchi, W. L. Soong, and F. Cupertino, *The Rediscovery of Synchronous Reluctance and Ferrite Permanent Magnet Motors: Tutorial Course Notes*. Springer, Apr. 2016.

[12] J. Lee, Y.-C. Kwon, and S.-K. Sul, "Experimental Identification of IPMSM Flux-Linkage Considering Spatial Harmonics for High-Accuracy Simulation of IPMSM Drives," in *2018 IEEE Energy Conversion Congress and Exposition (ECCE)*, Sep. 2018, pp. 5804–5809, iSSN: 2329-3748.

[13] "Gridded data interpolation - MATLAB - MathWorks United Kingdom." [Online]. Available: https://uk.mathworks.com/help/matlab/ref/griddedinterpolant.html

[14] "Interpolate 2-D or 3-D scattered data - MATLAB - MathWorks United Kingdom." [Online]. Available: https://uk.mathworks.com/help/matlab/ref/scatteredinterpolant.html

[15] S. Stipetic, D. Zarko, and M. Popescu, "Ultra-fast axial and radial scaling of synchronous permanent magnet machines," *IET Electric Power Applications*, vol. 10, no. 7, pp. 658–666, Aug. 2016. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1049/iet-epa.2016.0014

[16] M.-M. Koo, J.-Y. Choi, K. Hong, and K. Lee, "Comparative Analysis of Eddy-Current Loss in Permanent Magnet Synchronous Machine Considering PM Shape and Skew Effect Using 3-D FEA," *IEEE Transactions on Magnetics*, vol. 51, no. 11, pp. 1–4, Nov. 2015.

[17] R. Islam, I. Husain, A. Fardoun, and K. McLaughlin, "Permanent-Magnet Synchronous Motor Magnet Designs With Skewing for Torque Ripple and Cogging Torque Reduction," *IEEE Transactions on Industry Applications*, vol. 45, no. 1, pp. 152–160, Jan. 2009.

[18] A. Varatharajan, D. Brunelli, S. Ferrari, P. Pescetto, and G. Pellegrino, "syreDrive: Automated Sensorless Control Code Generation for Synchronous Reluctance Motor Drives," in *2021 IEEE Workshop on Electrical Machines Design, Control and Diagnosis (WEMDCD)*, Apr. 2021, pp. 192–197.