

Magnetic Modelling of Synchronous Reluctance and Internal Permanent Magnet Motors Using Radial Basis Function Networks

L. Ortombina, F. Tinazzi, *Member, IEEE*, and M. Zigliotto, *Member, IEEE*

Abstract—The general trend towards more-intelligent energy-aware ac drives is driving the development of new motor topologies and advanced model-based control techniques. Among the candidates, pure reluctance and anisotropic permanent magnet motors are gaining popularity, despite their complex structure. The availability of accurate mathematical models that describe these motors is essential to the design of any model-based advanced control. This paper focuses on the relations between currents and flux linkages, which are obtained through innovative radial-basis function neural networks. These special drive-oriented neural networks take as inputs the motor voltages and currents, returning as output the motor flux linkages, inclusive of any nonlinearity and cross-coupling effect. The theoretical foundations of the radial basis function networks, the design hints and a commented series of experimental results on a real laboratory prototype are included in the paper. The simple structure of the neural network fits for implementation on standard drives. The online training and tracking will be the next steps in FPGA-based control systems.

Index Terms—Permanent magnet motors, reluctance motor, artificial neural networks, magnetic flux linkages.

NOMENCLATURE

Variables names convention:

- *variable with accent $\hat{}$* : estimated quantities
- *bold lowercase variable*: vectorial quantities

Symbols used:

$u_{d,q}(t)$	Stator voltages
$i_{d,q}(t)$	Stator currents
$\lambda_{d,q}(i_d, i_q)$	Magnetic flux linkages
$\tau(t)$	Electromagnetic torque
$e^{d,q}$	Voltage estimation errors
N_g	
K	Number of Gaussian functions
b_k	Proportional coefficient of the first layer
d_{\max}	Diameter of the quadratic training region
n_k, \mathbf{x}_k, a_k	Input, centre in (d, q) reference frame and output of the k^{th} Gaussian functions
$w_k^{d,q}$	Proportional coefficients of the second layer
M	Number of steady state training points

$E^{d,q}$	Quadratic cost functions
$\mathbf{J}_{d,q}(\mathbf{w}_h^{d,q})$	Jacobian matrices at the h^{th} iteration
\mathbf{I}	Identity matrix of size equal to $\mathbf{J}_{d,q}(\mathbf{w}_h^{d,q})$
$\mu_h^{d,q}$	Coefficient to make matrices invertible
$\varepsilon_N^{d,q}$	Normalised flux estimation error

The time dependence (t) is omitted in the rest of the paper (unless otherwise specified) for the sake of more compact equations.

I. INTRODUCTION

Permanent Magnet (PM) synchronous motors suffer of the heavy ecological footprint of rare earth materials and their oscillating prices. The research is focusing on a drastic reduction of the use of PM, to obtain the best balance between the reluctance and the PM contributes to the electromagnetic torque. This is done through the investigation of several structure alternatives [1], [2]. The best candidates for the substitution in many applications are the synchronous reluctance (SynR) and internal permanent magnet (IPM) motors. The marked anisotropy of both SynR and IPM rotors is a distinctive advantage and major reason for a choice, since it also enables the sensorless control down to zero-speed.

To fully exploit the motor anisotropy, advanced control techniques as model reference adaptive control, maximum torque-per-ampere (MTPA), maximum torque-per-voltage (MTPV), sensorless control and model predictive control [3]–[5] need an accurate magnetic model of the motors. Actually, both SynR and IPM motors lack of the large equivalent airgap of surface-mounted PM motors and their magnetic circuits suffer of cross-coupling and saturation effects. The magnetic model becomes a non-linear set of relations among currents and flux linkages, usually expressed as two-input maps in a (d, q) reference frame synchronous to the rotor.

The key point in the choice of the model is the complexity of the system, which can be either simplified (losing accuracy) or tabulate (losing resolution). The finite element analysis (FEA) is a first viable way [6], even if it is quite sensitive to uncertainties in mechanical dimensions and material properties.

A simpler but not trivial alternative is based on test-bench measurements. The methods can be classified according to the speed at which the measurements are carried on. A first group computes the flux linkages by voltage integration in the (d, q) reference frame (at standstill) taking advantage of

the zeroing of the cross-coupling terms [7]–[9]. For example, [7] commendably includes the evaluation of the iron losses in the injection-based algorithm, investigating the behaviour of the differential inductances at high-frequency. The methods [9] and [8] apply a short sequence of bipolar voltage pulses to estimate the magnetic maps at standstill. The first one proposes a new flux saturation approximating function, which require the identification of multiple sets of constants to account for cross-coupling effect. The second one includes an empirical choice of some exponents in the mathematical approximation functions, that are borrowed by [10]. It is worth to note that in both cases the injection of a voltage excitation forces the use of a reduced DC bus voltage, which is not an industry standard. As a general remark, the common flaw of standstill methods is that they well suit for SynR motors, while they fail in estimating the PM flux linkage, when present.

As an alternative, two dynamic methods have been proposed in literature. In [11] the magnetic model is obtained by evaluating of cross-coupling voltage terms in a steady-state condition. The well-known influence of the phase resistance is smoothed by exploiting the flux linkage maps symmetry. The identification is quite accurate and for this reason they are often taken as a benchmark for comparisons. The only disadvantage is represented by the discrete output (look-up table, LUT), which poses problems of interpolation and differentiability. Alternatively, the magnetic model can be derived from accelerating and braking the rotor, at no load [12]–[14]. A fixed-current condition in the synchronous reference frame corresponds to a steady torque value that forces a linear speed ramp, for a mainly inertial load. During the acceleration, also the flux linkage vector is steady and it can be derived from the back-EMF estimate. The accuracy may be undermined by the iron losses, which change during the speed ramp. Also the speed increase may be a problem, since at no load and under a constant torque it can rapidly overcome the safety limit.

The present paper illustrates the study and the implementation of an accurate continuous 2D magnetic model of any IPM and SynR motor from a different point of view. The inherent complexity of the interaction between motor structure, currents and magnetic fluxes is considered as an ideal candidate for a “black-box” modelling. To this aim, artificial neural networks (ANNs) were selected as a powerful tool for mapping unknown non-linear relations [15]–[17]. The ANN structure of the proposed model is a simple 2-layer radial basis function (RBF) network, which is mainly composed by a set of bi-dimensional interconnected Gaussian functions [18]. The weighted sum of their output realises the non-linear map between the input (the current space vector) and the output (the motor flux linkages). To dispel the widespread belief that ANNs are “inappropriate” for their complexity, the paper aims at proving that the proposed solution well fits for implementation in standard drives, bringing inherent advantages.

In the paper, the theoretical background is presented in Sect. II, while the details of the RBF network its training algorithm are detailed in Sect. III. The practical implementation and validation are reported in Sect. V.

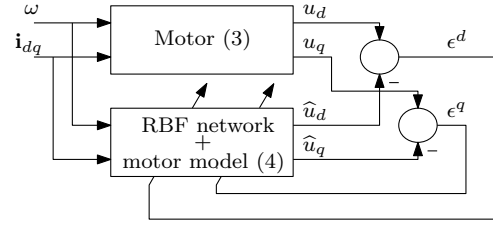


Fig. 1. RBF network training scheme.

II. THEORETICAL BACKGROUND

The voltage balance equations of a synchronous motor in the reference frame (d, q) fixed to the rotor are the following:

$$\begin{aligned} u_d(t) &= R_s i_d(t) + \frac{\partial \lambda_d(i_d(t), i_q(t))}{\partial t} - \omega_{me} \lambda_q(i_d(t), i_q(t)) \\ u_q(t) &= R_s i_q(t) + \frac{\partial \lambda_q(i_d(t), i_q(t))}{\partial t} + \omega_{me} \lambda_d(i_d(t), i_q(t)) \end{aligned} \quad (1)$$

and the electromagnetic torque delivered by the motor is:

$$\tau(t) = \frac{3}{2} p (\lambda_d(i_d, i_q) i_q - \lambda_q(i_d, i_q) i_d) \quad (2)$$

where for this time the dependence of both flux linkages on currents is made explicit. This paper proposes a new identification procedure that returns the (d, q) magnetic model of the motor, by the accurate measurement of phase currents, voltages and motor speed. The data are acquired at steady state, so that the equations (1) lose their derivative terms:

$$\begin{aligned} u_d &= R_s i_d - \omega_{me} \lambda_q(i_d, i_q) \\ u_q &= R_s i_q + \omega_{me} \lambda_d(i_d, i_q) \end{aligned} \quad (3)$$

The estimated voltages \hat{u}_d and \hat{u}_q , necessary for the comparison with the measured ones, are obtained as:

$$\begin{aligned} \hat{u}_d &= R_s i_d - \omega_{me} \hat{\lambda}_q(i_d, i_q) \\ \hat{u}_q &= R_s i_q + \omega_{me} \hat{\lambda}_d(i_d, i_q) \end{aligned} \quad (4)$$

where $\hat{\lambda}_d(i_d, i_q)$ and $\hat{\lambda}_q(i_d, i_q)$ are the estimated flux linkages that represents the output of the RBF network. The errors ϵ^d and ϵ^q are calculated as:

$$\begin{aligned} \epsilon^d &= u_d - \hat{u}_d = u_d - R_s i_d + \omega_{me} \hat{\lambda}_q \\ \epsilon^q &= u_q - \hat{u}_q = u_q - R_s i_q - \omega_{me} \hat{\lambda}_d \end{aligned} \quad (5)$$

Provided that voltages, currents, speed and resistance are known, the zeroing of both errors will result in the fulfilment of the equalities:

$$\lambda_d(i_d, i_q) = \hat{\lambda}_d(i_d, i_q) \quad \lambda_q(i_d, i_q) = \hat{\lambda}_q(i_d, i_q) \quad (6)$$

The block schematic of the training procedure is reported in Fig. 1. It recalls a model reference adaptive system, in which the adaptive model is represented by a combination of a motor model and the RBF network. The training is based on the voltage estimation errors (5) and the parameters adaptation mechanism of the RBF network will be described in Sect. III.

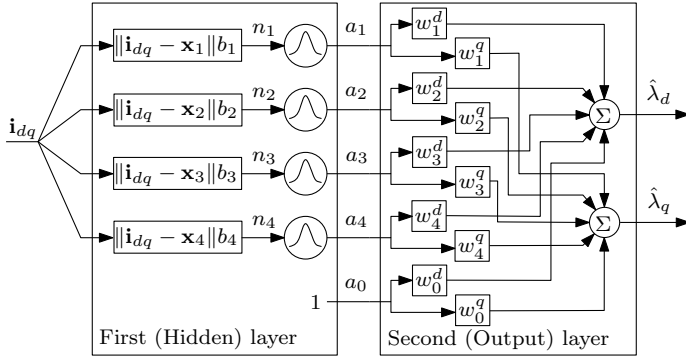


Fig. 2. Radial basis function network.

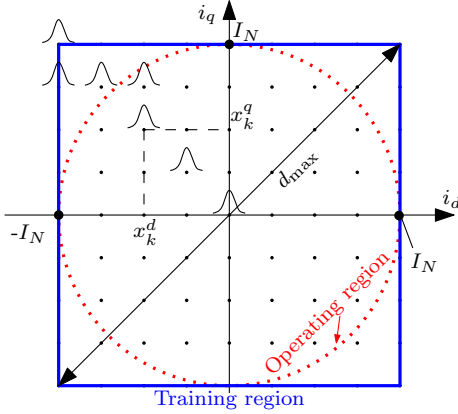


Fig. 3. Gaussian centres (dots) in the \mathbb{R}^2 input plane.

III. RBF NETWORK FUNDAMENTALS AND DESIGN

In principle, artificial neural networks can be considered as universal function approximators. There exist several types, that differ for topology and learning process. The choice must be application-oriented. This paper proposes a radial basis function neural network, which enables an easy implementation and, in perspective, an efficient online training and updating. The two-layer structure of the RBF network can be described with the aid of Fig. 2.

A. First layer

The hidden layer of a conventional multilayer perceptrons neural network (MLP-NN) is usually made of sigmoid functions [19], which are *global functions*, in the sense that their output is not zero even when the net input tends to infinity. Conversely, this paper proposes the innovative adoption of Gaussian functions in the hidden layer, in place of the sigmoids. These functions present a *local* characteristic, which means that the output is close to zero if inputs are conveniently far from the centre of the Gaussian. To understand the scope of the choice, it is worth to consider that, in general, the MLP-NN creates a distributed representation for each input, with the hidden layer functions collaborating and overlapping. Since in a RBF network each Gaussian function is active only around its centre, for any given input only few Gaussian functions will be active at a time. Such *local* property is interesting in perspective, because it facilitates the development of online

RBF updating algorithms at steady state. While being out of the scopes of the present work, it will be one of the next research steps.

As indicated in Fig. 2, the input to the RBF network is the phase current space vector \mathbf{i}_{dq} , defined within a circular region in the \mathbb{R}^2 plane (Fig. 3). For an easier implementation, the RBF network training region was extended to the square that circumscribes the circumference. Each dot in Fig. 3 represents the centre of a Gaussian function. The layout of the centres is a degree of freedom in the design of the RBF network. A comprehensive analysis is given in [20], that proposes either a fixed layout, with random positioning of the centres, or adaptive RBF centres. The conclusion is that the non-linear optimisation of the first layer parameters is beneficial only when a minimal network is required to solve a given problem. This is achieved at the cost of an overall increased complexity of the system.

For the present work (oriented to simple ac drives hardware) it has been found that a regular spacing of the RBF centres is a satisfactory solution. The lower the number of Gaussian functions, the lighter the implementation. To the authors experience, a good trade-off consists in spacing the Gaussian centres of about the 25% of the nominal current, in both axes directions. The result are $N_g \times N_g$ functions, with $N_g = 9$. As regards the parameters b_k (Fig. 2), a good guess is suggested in [19]:

$$b_k = \frac{\sqrt{K}}{2d_{\max}} \quad (7)$$

where $K = N_g^2$ is the number of Gaussian functions and d_{\max} is the maximum distance as defined in Fig. 3.

The hidden layer computes the Euclidean distance between each Gaussian centre $\mathbf{x}_k = (x_k^d + jx_k^q)$ and the input vector $\mathbf{i}_{dq} = i_d + ji_q$. The result is then multiplied by the coefficients b_k and sent as input to the Gaussian functions:

$$n_k = \|\mathbf{i}_{dq} - \mathbf{x}_k\| b_k \quad k = 1 \dots K \quad (8)$$

B. Second layer

The second layer is the conventional one adopted in most of MLP-NN. Only four Gaussian functions are drawn in Fig. 2, but actually many others are present and cover with the due resolution the whole input range.

The hidden layer Gaussian outputs are $a_k = e^{-n_k^2}$. The flux linkages space vector estimate $\hat{\lambda}_{dq}$ can be expressed as linear combinations of those outputs, weighted by the coefficients $w_k^{d,q}$ (Fig. 2):

$$\hat{\lambda}_d = \sum_{k=0}^K w_k^d e^{-n_k^2} \quad \hat{\lambda}_q = \sum_{k=0}^K w_k^q e^{-n_k^2} \quad (9)$$

The RBF weights $w_k^{d,q}$ will be the object of the training procedure described in Sec. IV.

In real-time applications the computational burden of (9) exceeds that of LUT-based methods. On the other hand, one may appreciate the advantages brought about by the continuity of the proposed functions-based model and some practical tricks can be adopted to smooth the gap. In the present work, the exponentials of (9) were approximated by means of

polynomial functions of fifth order. Their computation required about the 25% of the control cycle, which was of 100 μ s, so that enough time was left for the remaining control tasks.

IV. RBF NETWORK TRAINING

The RBF weights training procedure consists of two steps. It starts by the acquisition of voltages, currents and speed measurements in multiple steady state conditions, followed by an offline training.

A. Data acquisition

In order to minimise the errors due to temperature variations, the motor is first warmed up to the thermal equilibrium. Two current references, namely i_d^* and i_q^* , are imposed to the motor under test, which is driven at fixed speed by a second independent motor unit. The speed reference comes out as a design trade-off. On one hand, the speed should be low to reduce the influence of iron losses, which drain part of the stator current deputed to the flux production, [21]. On the other hand, there is the request of good signal-to-noise ratio for the voltage measurement, which gets better at higher speed. The experimental RBF training was performed at 100 rpm.

The region of Fig. 3 was gridded to get current references equally spaced-out from each other. The total number of successive steady state training points was:

$$M = (2I_N/\Delta I + 1)^2 \quad (10)$$

where $\Delta I = I_N/10$ is the result of a design trade-off between resolution and data storage space. For each of the M steady state working points, the voltage and currents were sampled every $T_c = 100 \mu$ s and averaged over a complete mechanical revolution. This smooths possible disturbances occurring at either the electrical or mechanical frequency, caused by mechanical and winding asymmetries.

At the end of the first step, M reliable samples of the vectors \mathbf{u}_{dq} , \mathbf{i}_{dq} and the speed ω_{me} are available, for the subsequent training of the RBF network.

B. Offline network training

Two different sets of weights $\mathbf{w}^d = [w_1^d \dots w_K^d]$ and $\mathbf{w}^q = [w_1^q \dots w_K^q]$ are used for each flux linkage estimate, as shown in Fig. 2. During the training, they were iteratively adjusted by means of the Levenberg-Marquardt (LM) algorithm, chosen for its documented property of fast convergence, even from a rather wrong initial guess [19]. Essentially, LM is a damped least-squares method that calculates the weights with the goal of minimising the two quadratic cost functions defined as:

$$\begin{aligned} E^d(\mathbf{w}^d) &= \frac{1}{2} \sum_{i=1}^M (\epsilon_i^d)^2 = \frac{1}{2} (\boldsymbol{\epsilon}^d)^T \boldsymbol{\epsilon}^d \\ E^q(\mathbf{w}^d) &= \frac{1}{2} \sum_{i=1}^M (\epsilon_i^q)^2 = \frac{1}{2} (\boldsymbol{\epsilon}^q)^T \boldsymbol{\epsilon}^q \end{aligned} \quad (11)$$

As shown in Fig. 1, in this case the inputs to the LM algorithm were the voltage estimation errors $\boldsymbol{\epsilon}^d = [\epsilon_1^d, \epsilon_2^d, \dots, \epsilon_M^d]$

and $\boldsymbol{\epsilon}^q = [\epsilon_1^q, \epsilon_2^q, \dots, \epsilon_M^q]$. These errors were obtained from the measurements, according to (5) and can be both initialised to zero.

It is worth to note that the cost function E^d depends on $\hat{\lambda}_q$, due to the motional cross-coupling terms in (5). Therefore, E^d is function of \mathbf{w}^q and vice versa for E^q . This cross-dependence has been made explicit in (11).

Assuming that \mathbf{w}_h^d and \mathbf{w}_h^q are the weight sets that minimise the cost functions E^d, E^q at the h -th iteration of the LM algorithm, let the Jacobian matrix $\mathbf{J}_d(\mathbf{w}_h^d)$ of the vector-valued function $\boldsymbol{\epsilon}^q(\mathbf{w}_h^d)$ be defined as:

$$\mathbf{J}_d(\mathbf{w}_h^d) = \begin{bmatrix} \frac{\partial \epsilon_1^q(\mathbf{w}_h^d)}{\partial w_1^d} & \frac{\partial \epsilon_1^q(\mathbf{w}_h^d)}{\partial w_2^d} & \dots & \frac{\partial \epsilon_1^q(\mathbf{w}_h^d)}{\partial w_K^d} \\ \frac{\partial \epsilon_2^q(\mathbf{w}_h^d)}{\partial w_1^d} & \frac{\partial \epsilon_2^q(\mathbf{w}_h^d)}{\partial w_2^d} & \dots & \frac{\partial \epsilon_2^q(\mathbf{w}_h^d)}{\partial w_K^d} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \epsilon_M^q(\mathbf{w}_h^d)}{\partial w_1^d} & \frac{\partial \epsilon_M^q(\mathbf{w}_h^d)}{\partial w_2^d} & \dots & \frac{\partial \epsilon_M^q(\mathbf{w}_h^d)}{\partial w_K^d} \end{bmatrix} \quad (12)$$

A similar definition holds for $\mathbf{J}_q(\mathbf{w}_h^q)$. The M current vector measurements are proposed in sequence to the network input, and the related estimation errors are computed accordingly. Then the LM weights updating laws are applied:

$$\begin{aligned} \mathbf{w}_{h+1}^d &= \mathbf{w}_h^d - [\mathbf{J}_d^T \mathbf{J}_d + \mu_h \mathbf{I}]^{-1} \mathbf{J}_d^T \boldsymbol{\epsilon}_h^q \\ \mathbf{w}_{h+1}^q &= \mathbf{w}_h^q - [\mathbf{J}_q^T \mathbf{J}_q + \mu_h \mathbf{I}]^{-1} \mathbf{J}_q^T \boldsymbol{\epsilon}_h^d \end{aligned} \quad (13)$$

As a distinctive feature of the LM algorithm, the coefficient μ_h is added to make the two matrices $[\mathbf{J}_{d,q}^T \mathbf{J}_{d,q} + \mu_h \mathbf{I}]$ certainly invertible.

Provided that the data set is well conditioned (i.e. the measurements are well spaced in the training region), the coefficient μ_h can be set very close to zero, speeding up the training process. At least, when $\mu_h = 0$, the search of the optimal weight vectors is performed in a single iteration only.

Each element of the matrix (12) can be made explicit by using (5). For example, for the first element it holds:

$$\frac{\partial \epsilon_1^q(\mathbf{w}^d)}{\partial w_1^d} = \frac{\partial (u_q - R_s i_q - \omega_{me} \hat{\lambda}_d(\mathbf{w}^d))}{\partial w_1^d} \quad (14)$$

It is worth to note that the stator resistance R_s is supposed to be known, and u_q, i_q e ω_{me} are measurements, independent from w_1^d . With reference to (9) and the symbols used in Fig. 2, equation (14) is simplified as follows:

$$\frac{\partial \epsilon_1^q}{\partial w_1^d} = -\omega_{me} \frac{\partial \hat{\lambda}_d}{\partial w_1^d} = -\omega_{me} \underbrace{\frac{\partial \left(\sum_{k=1}^K a_k w_k^d + w_b^d \right)}{\partial w_1^d}}_{a_1} \quad (15)$$

In the light of (15), the computation of $\mathbf{J}_d(\mathbf{w}_h^d)$ and $\mathbf{J}_q(\mathbf{w}_h^q)$ is trivial, since the a_k terms are already available (because they are used to compute the voltage estimation errors $\boldsymbol{\epsilon}_{d,q}$).

It is worth to note that $\mathbf{J}_{d,q}$ are $M \times K$ matrices, which are the number of measurements and the number of Gaussian functions, respectively. Their size increases rapidly, with heavy implications on the eventual online version of the algorithm.

TABLE I
SYNCHRONOUS MOTORS NAMEPLATE DATA

	IPM	SynR
Nominal current (I_N)	4.2 A	4 A
Nominal speed ($\omega_{m,N}$)	3000 rpm	1500 rpm
Nominal Torque (τ_N)	4.5 N m	5.5 N m
Pole pairs (p)	2	2
Stator resistance (R_s)	2.73 Ω	4.76 Ω
L_d inductance (unsat.)	21 mH	380 mH
L_q inductance (unsat.)	114 mH	85 mH
PM flux linkage (Λ_{mg})	0.23 V s	–

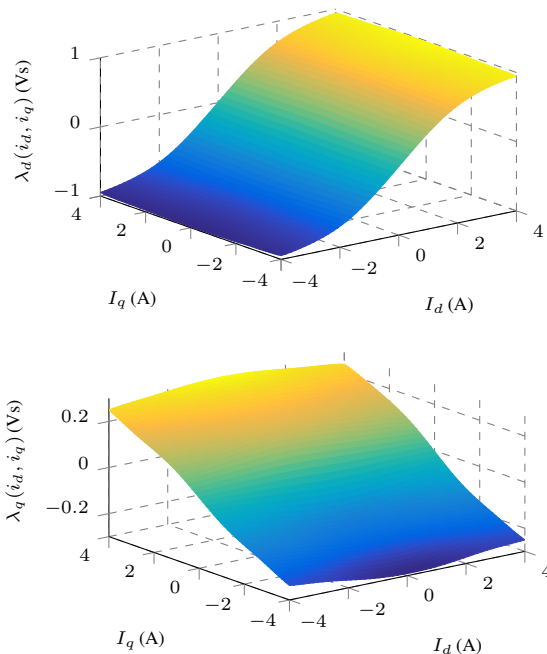


Fig. 4. 3D magnetic maps of the SynR motor (experimental).

With the actual standard ac drives computational power, and as long the LM algorithm is maintained in this form, the RBF training can be performed offline only. However, it is believed that the growing popularity of very fast devices, as field programmable gate array (FPGA), along with some mathematical optimisation, will rapidly lead to a complete online RBF network training.

V. EXPERIMENTAL RESULTS

The experiments were performed on both an IPM motor and a SynR motor, whose parameters are reported in TABLE I. The motors under test were fed by a two-level three-phase IGBT voltage inverter, controlled by a fast control prototyping system featuring and programmed in C-language. The motor under test was current-controlled and the current references were generated as explained in Sec. IV. The motor phase-to-phase voltages (two out of three) were measured with a custom digital measurement system, based on fast PWM signal oversampling, post-processed by a dedicate FPGA chip [22].

The stator resistance R_s has to be known with good accuracy, to compute (11). A precise online tracking algorithm, based on a modified version of that proposed in [23], was

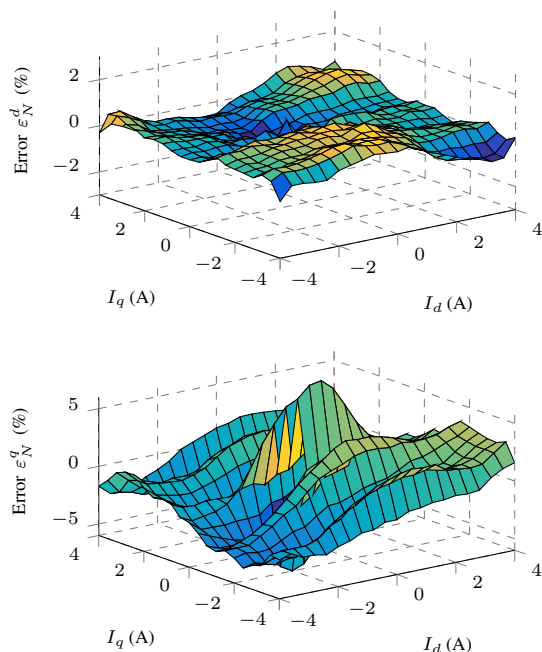


Fig. 5. SynR motor flux linkages mismatch, proposed method vs. estimate obtained as in [11].

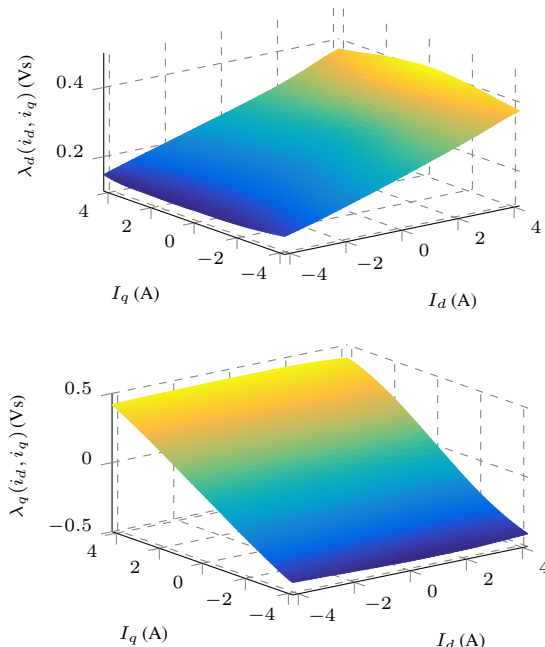


Fig. 6. 3D magnetic maps of the IPM motor (experimental).

implemented, achieving a tracking accuracy within the 3% of the nominal value at 25 °C in the whole operating range of the drive.

In order to evaluate the proposed magnetic mapping technique, the results have been compared with those obtained by an offline benchmark method [11]. Let the normalised error be defined as

$$\varepsilon_N^{d,q} = \frac{\hat{\lambda}_{d,q} - \lambda_{d,q}}{\lambda_{d,q \max}} \cdot 100 \quad (16)$$

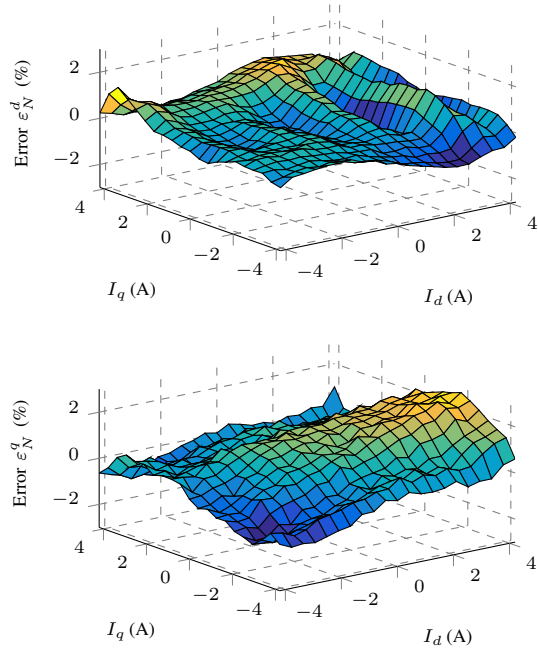


Fig. 7. IPM motor flux linkages mismatch, proposed method vs. estimate obtained as in [11].

where $\hat{\lambda}_{d,q}$ are the RBF network estimates, $\lambda_{d,q}$ are the “true” values obtained with the method proposed in [11] and $\lambda_{d,q \max}$ are the two maximum values of flux linkages along each axis.

The magnetic maps of the SynR motor, generated by the first trained RBF network, are reported in Fig. 4. The normalised errors with respect to the cited benchmark are reported in Fig. 5. Similarly, the magnetic maps of the IPM motor and the normalised errors are shown in Fig. 6 and Fig. 7, respectively.

The error magnitude remains almost within $\pm 3\%$ in both cases. In particular, the normalised d -axis error is always within $\pm 1\%$. At very low currents, the q -axis error of the SynR motor increases up to 5%. But the attribution of the whole error to the RBF estimate is probably unfair, since in that region the variance of the measurements errors could play a major role. As a countermeasure, a more accurate result could be obtained by increasing the number of neurons (Gaussian functions) in the region of interest. This can always be done, as soon as the regular distribution proposed in Fig. 3 proves inadequate. Of course, any increase of K or M brings along a not negligible memory consumption or computation time.

For the sake of comparison, the curves obtained with the two extreme cross-coupling conditions, in which the other current is either null, or close to its nominal value, are reported in Fig. 8. The match between the two models is very accurate in any operating condition. Being linear combinations of Gaussian functions, flux linkages are continuous function of the currents (black lines), while the magnetic maps obtained by the reference method [11] are discrete points (red circles).

The continuity and the derivability of the flux linkages with respect to currents is a major feature of the proposed technique. The computation of the differential inductances is a significant example in this sense. They can be obtained by differentiating (9) with respect to both i_d and i_q . Conversely,

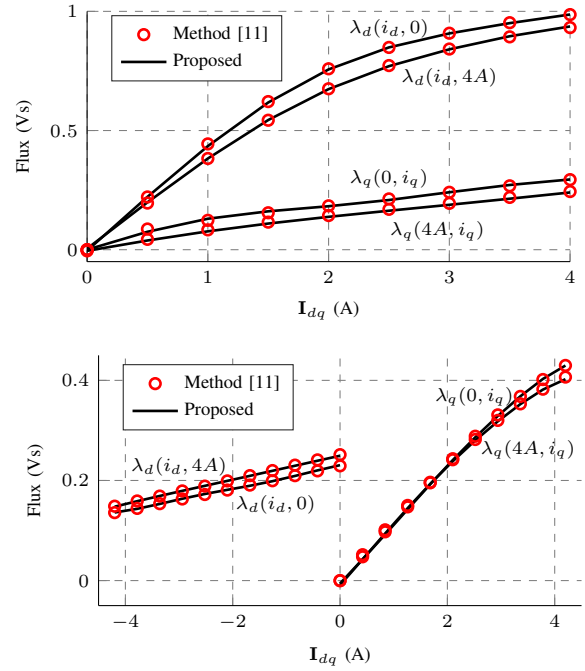


Fig. 8. 2D magnetic maps (upper: SynR lower: IPM). For each axis, the flux linkages are computed either with maximum or without cross-coupling effect.

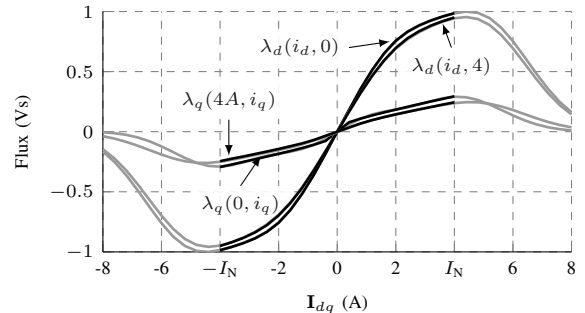


Fig. 9. Flux linkages estimation up to the nominal current (black) and up to twice the nominal current (grey). The RBF network was trained up to the nominal current value only.

the discrete nature of the LUT-based model forces the computation of the difference quotient, which returns the piecewise constant inductance. The results for both motors are reported in Fig. 10.

For safety reasons, in the present work the RBF network was trained up to the nominal motor currents only. To investigate the behaviour of the RBF network outside the training region, a simulation with currents up to twice the nominal value was performed. The results are reported in Fig. 9. As expected, the estimated flux linkages drop quite rapidly as the current exceeds the boundary of the training region (Fig. 3). This tendency is amplified by the *local* property of the network (Sec. III-A), which ultimately prevents each Gaussian function to contribute to the output only when the input is outside the working region. Of course, the working region (and the training) can be extended beyond the nominal current values, if requested by the application. In LUT-based models, this is

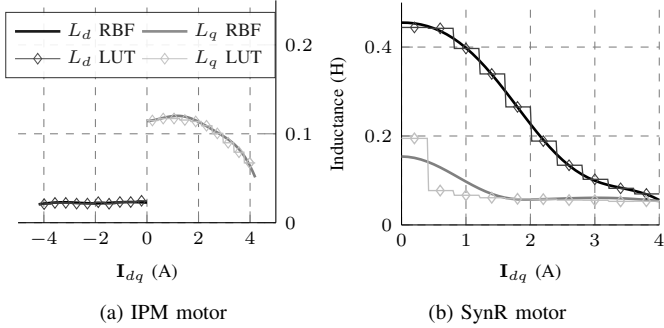


Fig. 10. Differential inductances as function of their respective axis current (while the current in the other axis is taken equal to zero).

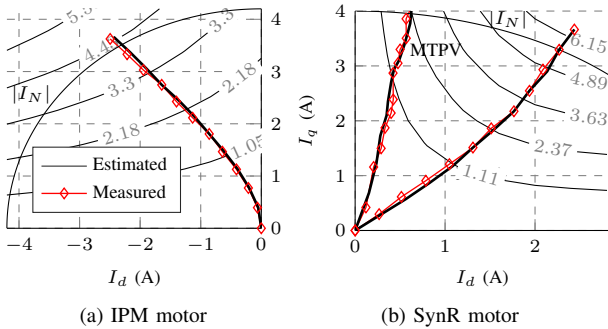


Fig. 11. Assessment of accuracy through the comparison of MTPA and MTPV curves (for the SynR motor, right). Since $\Lambda_{mg}/L_d > I_N$, for the IPM motor only the MTPA curve is reported.

equivalent to enlarge the table size.

The advantages of the RBF-based magnetic model be summarised as follows:

- it gives a continuous approximation of the inherent non-linear magnetic relation, including cross-coupling effect;
- it fits for online model-predictive control algorithms that require also the flux linkage derivatives;
- in perspective, the special RBF network structure smooths the ways for online training and tracking.

The last point has positive consequences also in diagnostics, since updated flux linkages could help in detecting faulty operations. An autonomous self-calibrating model would also match the call for “more on-board intelligence” in the ac drives of the next generation.

A. Assessment of model accuracy through MTPA curves

An assessment of the precision of the RBF-based model can be obtained indirectly by forcing both IPM and SynR motor drives to follow the MTPA and MPTV curves, and then comparing the resulting trajectories with the ones obtained by using the reference LUT. The two-step accuracy test was performed for both motors of TABLE I and the results are reported in Fig. 11a and Fig. 11b, respectively.

First, the real MTPA curve was obtained by measurements. In particular, the synchronous motor was speed-controlled at constant speed against a variable load torque. For different torque levels ranging from 0 to τ_N the phase of the reference

current space vector was swept to seek the one relative to the minimum-amplitude vector. The result was a collection of MTPA points that can be connected to form the “measured” MTPA curve in Fig. 11. In the experiment, the shaft torque was measured by a torquemeter. Second, the ideal MTPA curve was computed by exploiting the electromagnetic torque (2) and the magnetic model (9) obtained by the RBF-based model (Fig. 4 and Fig. 6). The mathematical details are reported in Appendix A. The same assessment methodology were used for the comparison of the MTPV curves, Fig. 11b. One should note that since $\Lambda_{mg}/L_d > I_N$, it was not possible to calculate the MTPV curve for the IPM motor.

Actually, the superposition is almost perfect, testifying the high accuracy of the proposed technique. The slight mismatch between the IPM motor curves at higher currents can be ascribed to the flatness of the torque vs. current surface for that motor. It reduces the sensitivity of MTPA algorithm and may induce some imprecision in the measured curve. Precisely because of the flatness of the surface, the possible error has little influence on performance. Last, but not least, it is worth noting that (20) can also be considered as the rough starting point for the future online MTPA exact computation, provided that it is properly worked out and simplified.

VI. CONCLUSIONS

The study and application of RBF networks to magnetic mapping of any synchronous motor is the original contribute of this work and it yields some interesting advantages. A smooth and precise d, q flux linkages estimation enables the use of the model in any model-based control system. In particular, algorithms that require integral/derivative operations on flux linkages can take great benefits.

The network training procedure is kept as easy as possible, but it still requires the inversion of a matrix, which cannot be embedded in a real-time implementation yet. But the structure of the selected RBF neural network is open to further simplifications. In short times, this should lead to the online implementation not only of the commissioning process, but also of the RBF network training and flux-linkages tracking algorithm.

The RBF network has a so-called *local* property. This means that the synapses weights will be updated only in the portion of plane around the present steady state working point. This is important to prevent the updating of a part of the map from interfering in the remaining working space. Future research developments will include the development of the continuous online tracking of the magnetic model via RBF network update and the transposition of the whole algorithm in a fast parallel FPGA circuit.

APPENDIX A

At steady state, the current space vector can be expressed in the polar coordinates (I, ϑ) as $\mathbf{i}_{dq} = I e^{j\vartheta}$. For any $0 \leq I \leq I_N$ the MTPA trajectory fulfils the equation

$$\frac{\partial \tau(I, \vartheta)}{\partial \vartheta} = \frac{3}{2} p I \frac{\partial (\hat{\lambda}_d \sin \vartheta - \hat{\lambda}_q \cos \vartheta)}{\partial \vartheta} = 0 \quad (17)$$

Thanks to the continuity of the magnetic map expressions, it is possible to further expand the derivative (17). By using (8) and (9), and if one sets:

$$w_d = \sum_{k=0}^K w_k^d \frac{\partial a_k}{\partial \vartheta} \quad w_q = \sum_{k=0}^K w_k^q \frac{\partial a_k}{\partial \vartheta} \quad (18)$$

where

$$\frac{\partial a_k}{\partial \vartheta} = 2a_k b_k^2 I(x_k^q \cos(\vartheta) - x_k^d \sin(\vartheta)) \quad (19)$$

the MTPA condition (17) becomes:

$$\begin{aligned} \frac{\partial \tau(I, \vartheta)}{\partial \vartheta} &= \frac{3}{2} p I [w_d \sin(\vartheta) - w_q \cos(\vartheta)] + \\ &\sum_{k=0}^K w_k^d a_k \cos(\vartheta) + \sum_{k=0}^K w_k^q a_k \sin(\vartheta) = 0 \end{aligned} \quad (20)$$

The transcendent equation (20) was solved numerically offline for an appropriate number of points, to get the bold “estimated” MTPA curve reported in Fig. 11.

REFERENCES

- [1] E. Carraro, M. Morandini, and N. Bianchi, “Traction PMASR motor optimization according to a given driving cycle,” *IEEE Trans. Ind. Appl.*, vol. 52, no. 1, pp. 209–216, Jan. 2016.
- [2] T. Kato, M. Minowa, H. Hijikata, K. Akatsu, and R. Lorenz, “Design methodology for variable leakage flux IPM for automobile traction drives,” *IEEE Trans. Ind. Appl.*, vol. 51, no. 5, pp. 3811–3821, Sep. 2015.
- [3] Q. Liu and K. Hameyer, “High-performance adaptive torque control for an ipmsm with real-time mtpa operation,” *IEEE Trans. Energy Convers.*, vol. 32, no. 2, pp. 571–581, Jun. 2017.
- [4] R. Antonello, M. Carraro, L. Peretti, and M. Zigliotto, “Hierarchical scaled-states direct predictive control of synchronous reluctance motor drives,” *IEEE Trans. Ind. Electron.*, vol. 63, no. 8, pp. 5176–5185, Aug. 2016.
- [5] P. Guglielmi, M. Pastorelli, and A. Vagati, “Cross-saturation effects in IPM motors and related impact on sensorless control,” *IEEE Trans. Ind. Appl.*, vol. 42, no. 6, pp. 1516–1522, Nov. 2006.
- [6] G. Bramerdorfer, S. M. Winkler, M. Kommenda, G. Weidenholzer, S. Silber, G. Kronberger, M. Affenzeller, and W. Amrhein, “Using fe calculations and data-based system identification techniques to model the nonlinear behavior of pmsms,” *IEEE Trans. Ind. Electron.*, vol. 61, no. 11, pp. 6454–6462, Nov. 2014.
- [7] L. Peretti, P. Sandulescu, and G. Zanuso, “Self-commissioning of flux linkage curves of synchronous reluctance machines in quasi-standstill condition,” *IET Electric Power Applications*, vol. 9, no. 9, pp. 642–651, 2015.
- [8] M. Hinkkanen, P. Pescetto, E. Mölsä, S. E. Saarakkala, G. Pellegrino, and R. Bojoi, “Sensorless self-commissioning of synchronous reluctance motors at standstill without rotor locking,” *IEEE Trans. Ind. Appl.*, vol. 53, no. 3, pp. 2120–2129, May. 2017.
- [9] N. Bedetti, S. Calligaro, and R. Petrella, “Stand-still self-identification of flux characteristics for synchronous reluctance machines using novel saturation approximating function and multiple linear regression,” *IEEE Trans. Ind. Appl.*, vol. 52, no. 4, pp. 3083–3092, Jul. 2016.
- [10] Z. Qu, T. Tuovinen, and M. Hinkkanen, “Inclusion of magnetic saturation in dynamic models of synchronous reluctance motors,” in *ICEM*, pp. 994–1000, Sep. 2012.
- [11] E. Armando, R. Bojoi, P. Guglielmi, G. Pellegrino, and M. Pastorelli, “Experimental identification of the magnetic model of synchronous machines,” *IEEE Trans. Ind. Appl.*, vol. 49, no. 5, pp. 2116–2125, Sep. 2013.
- [12] G. Pellegrino, B. Boazzo, and T. M. Jahns, “Magnetic model self-identification for PM synchronous machine drives,” *IEEE Trans. Ind. Appl.*, vol. 51, no. 3, pp. 2246–2254, May. 2015.
- [13] S. Wiedemann, R. M. Kennel, S. Hall, and M. Alakla, “Dynamic testing characterization of a synchronous reluctance machine,” in *EVER*, pp. 1–7, Apr. 2016.
- [14] S. Hall, F. J. Marquez-Fernandez, and M. Alakula, “Dynamic magnetic model identification of permanent magnet synchronous machines,” *IEEE Trans. Energy Convers.*, vol. PP, no. 99, pp. 1–1, 2017.
- [15] S. Haykin, *Neural Network and Learning Machines*, 3rd. Prentice Hall, 2008.
- [16] L. Ortombina, F. Tinazzi, and M. Zigliotto, “Comprehensive magnetic modelling of internal PM synchronous motors through radial basis function networks,” in *IECON*, pp. 4319–4324, Oct. 2016.
- [17] K. Liu, Z. Q. Zhu, and D. A. Stone, “Parameter estimation for condition monitoring of pmsm stator winding and rotor permanent magnets,” *IEEE Trans. Ind. Electron.*, vol. 60, no. 12, pp. 5902–5913, Dec. 2013.
- [18] S. Haykin, *Neural Networks: A Comprehensive Foundation*. MacMillan Publishing Company, 1994.
- [19] M. Hagan, H. Demuth, M. Beale, and O. De Jesús, *Neural Network Design*, 2nd, 2014, 978-0-9717321-1-7. [Online]. Available: <http://hagan.okstate.edu/nnd.html>
- [20] D. Lowe, “Adaptive radial basis function nonlinearities, and the problem of generalisation,” in *Artificial Neural Networks, 1989., First IEE International Conference on*, pp. 171–175, Oct. 1989.
- [21] N. Urasaki, T. Senjyu, and K. Uezato, “A novel calculation method for iron loss resistance suitable in modeling permanent-magnet synchronous motors,” *IEEE Trans. Energy Convers.*, vol. 18, no. 1, pp. 41–47, Mar. 2003.
- [22] R. Antonello, F. Tinazzi, and M. Zigliotto, “Benefits of direct phase voltage measurement in the rotor initial position detection for permanent-magnet motor drives,” *IEEE Trans. Ind. Electron.*, vol. 62, no. 11, pp. 6719–6726, Nov. 2015.
- [23] G. Zanuso, L. Peretti, and P. Sandulescu, “Stator reference frame approach for DC injection-based stator resistance estimation in electric drives,” in *2015 IEEE PEDS*, pp. 867–872, Jun. 2015.



Ludovico Ortombina received the B.S. and M.S (cum laude) in mechatronics engineering from the University of Padova, Vicenza, Italy, in 2013 and 2015, respectively. He is currently working toward the Ph.D. degree in mechatronics engineering in the Department of Management and Engineering, University of Padova. His research interests include control and parameter estimation techniques for ac motors.



Fabio Tinazzi (M '16) received the B.S. and M.S., Ph.D. degree in Mechatronic Engineering from University of Padova in 2008, 2011 and 2015 respectively. Since February 2017, he is a Researcher at the Department of Management and Engineering, University of Padova, Vicenza, Italy. His main research interests include sensorless control and parameter estimation techniques for ac motors.



Mauro Zigliotto (M '98) is a native of Vicenza, Italy. He is full professor of Electrical Machines and Drives at University of Padova, Italy and head of the Electric Drives Laboratory in Vicenza, Italy. Advanced control strategies and self-commissioning for ac motors are Prof. Zigliotto's main research interests. He is the secretary of the IEEE IAS-IES-PELS North Italy Joint Chapter.