

Received March 18, 2022, accepted April 11, 2022, date of publication April 21, 2022, date of current version May 4, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3169149

Efficiency Map-Based PMSM Parameters Estimation Using Power-Oriented Modeling

DAVIDE TEBALDI 

Department of Engineering “Enzo Ferrari,” University of Modena and Reggio Emilia, 41125 Modena, Italy

e-mail: davide.tebaldi@unimore.it

ABSTRACT This paper deals with two subjects. The first one is the presentation of a Matlab App created for PMSMs (Permanent Magnet Synchronous Motors) parameters estimation, both in the electrical and in the mechanical energetic domains. The second one is the presentation of a generalized PMSM Simulink superblock equipped with a user-friendly interface allowing to select options and to input the model parameters. The Matlab App and the PMSM superblock can be easily interfaced with each other. Finally, two simulation case studies are reported for the PMSM superblock.

INDEX TERMS Parameters estimation, PMSM, simulation, system efficiency, system modeling.

I. INTRODUCTION

PMSMs are employed in a very large variety of industrial fields, including robotics, aerospace, road and rail electric vehicles [1] and renewable energy applications such as solar water pumping [2]. Out of all the advantages PMSMs provide, good dynamic performances, high power density and high efficiency are surely among the most interesting ones [3]–[5]. The PMSM has to be chosen in order to satisfy the requirements of the given application, e.g. desired speed and torque range [5] etc. Once the most suitable PMSM has been identified, the modeling step has to be addressed, which is essential to effectively describe the machine behavior. Different modeling approaches can be found in the literature. As an example, the PMSMs modeling is addressed in a d-q discrete-time reference frame in [6]. A modeling in the d-q frame is also employed in [7]–[10], whereas a fault diagnosis model based on a Siamese network is proposed in [11]. A fundamental advantage of model-based methods is the knowledge of the mathematical model of the system under consideration. This is not only important for an engineer to really understand the dynamics of the analyzed physical system, but it also allows to develop more effective control strategies which are based on the model knowledge. The PMSM models employed in all these works are very useful but lack of flexibility, i.e. they do not allow the user to *select options*, meaning whether to simulate in the three-phase static frame or the d-q frame or whether to account for the system mechanical dynamics or not, in order to perform lighter or heavier simulations. In [12] and [13], the Simulink blocks for interior and surface PMSM are presented. They allow a larger flexibility, but the inner structure of the model and the power flowing

between the electrical and mechanical parts of the machine cannot be easily understood, since the power sections are not highlighted. The physical systems modeling can also be effectively addressed using suitable energy-based graphical formalisms such as Power-Oriented Graphs (POG) [14], Bond-Graph (BG) [15] and Energetic Macroscopic Representation (EMR) [15]. In this work, the POG modeling technique is used [14]. The latter is a graphical formalism which is very useful for describing systems dynamic models, as it allows to build block schemes which are directly implementable in Simulink and it also allows to effectively control the system power flows [16]. In this paper, a PMSM Simulink superblock is proposed [17] `PMSM_Superblock`, offering the following interesting functionalities: 1) straightforward compatibility with the PMSM parameters estimation Matlab App [17] `Estimation_Matlab_App` proposed in this paper; 2) both speed and torque control are incorporated in the model; 3) the user can choose whether to compute the three-phase voltages and currents or not, leading to heavier or lighter simulations; 4) the user can choose if the rotor mechanical dynamics has to be included or not; 5) easy control and understanding of the power flows within the system.

The next important step to be addressed is PMSM parameters estimation, for which several approaches have been proposed in the literature. For example, a model reference adaptive system is proposed in [18], whereas a deterministic learning approach for identifying the system dynamics is addressed in [19]. A genetic algorithm for the search of the minimal point is studied in [20], whereas two offline and online estimation methods involving different system parameters are shown in [21] and [22]. Some other studies focus instead on the temperature estimation of PMSM, as shown in [23] where a Feedforward Neural Network is employed. In order to overcome the problem of parameters

The associate editor coordinating the review of this manuscript and approving it for publication was Claudia Raibulet.

estimation, efforts have been made to develop model-free control approaches for PMSMs, like the model-free predictive current control proposed in [24]. In this case, the concern is related to the current difference updating frequency in order to obtain a good performance. A compensated stator flux predictor is instead proposed in [25] in order to improve the prediction accuracy of the stator flux linkage vector. An exhaustive review of the current state of the art for PMSM offline and online estimation techniques is proposed in [5]. As far as online numerical methods are concerned, two examples are Recursive Least Square (RLS) and Extended Kalman Filter (EKF). Using RLS [26], a current injection-based method is proposed in [27], [28], whereas EKF is employed in [29].

In this paper, a Matlab App for PMSM parameters estimation [17] Estimation_Matlab_App, both in the electrical and in the mechanical energetic domains, is proposed. This can be potentially suitable for self-commissioning applications, and can also be potentially extended to be used online. Furthermore, the Matlab App and the PMSM superblock can be easily interfaced with each other.

The remainder of this paper is organized as follows. Sec. II addresses the PMSM modeling and efficiency analysis. Sec. III presents the user interface and parameters definition of the Matlab App for PMSM parameters estimation, whereas Sec. IV describes its operation and working principles. Sec. V introduces the PMSM Simulink superblock together with two simulation examples, while the conclusions are finally reported in Sec. VI.

II. PMSM MODELING AND EFFICIENCY

The state-space model of the PMSM in the d-q frame Σ_ω can be written as follows:

$$\underbrace{\begin{bmatrix} pL_{se} & 0 & 0 & 0 \\ 0 & pL_{se} & 0 & 0 \\ 0 & 0 & p(L_s - M_{s0}) & 0 \\ 0 & 0 & 0 & J_m \end{bmatrix}}_L \underbrace{\begin{bmatrix} \dot{\omega}_{I_d} \\ \dot{\omega}_{I_q} \\ \dot{\omega}_{I_{s3}} \\ \dot{\omega}_m \end{bmatrix}}_{\dot{x}} = \underbrace{\begin{bmatrix} pR_s & -p^2\omega_m L_{se} & 0 & \omega K_d \\ p^2\omega_m L_{se} & pR_s & 0 & \omega K_q \\ 0 & 0 & pR_s & \omega K_{s3} \\ -\omega K_d & -\omega K_q & -\omega K_{s3} & b_m \end{bmatrix}}_A \underbrace{\begin{bmatrix} \omega_{I_d} \\ \omega_{I_q} \\ \omega_{I_{s3}} \\ \omega_m \end{bmatrix}}_x + \underbrace{\begin{bmatrix} \omega V_d \\ \omega V_q \\ \omega V_{s3} \\ -\tau_e \end{bmatrix}}_u \quad (1)$$

where L and A are the system energy and power matrices and x and u are the system state and input vectors. Terms ωK_d , ωK_q , ωK_{s3} are the three components of the PMSM torque vector $\omega K_\tau(\theta)$, where θ is the electric angle, whereas ωI_d , ωI_q , ωI_{s3} and ωV_d , ωV_q , ωV_{s3} are the three components of the PMSM current and voltage vectors ωI_s and ωV_s . The meaning

TABLE 1. Meaning of the parameters in systems (1), (2) and (3).

p	Number of polar expansions;
R_s	Stator phase resistance ($p = 1$);
L_{se}	$L_s + \frac{1}{2}M_{s0}$;
L_s	Stator phase self inductance coefficient ($p = 1$);
M_{s0}	Maximum value of mutual inductance between stator phases;
b_m	Rotor linear friction coefficient;
ω_m	Motor angular velocity;
ωK_q	Motor torque constant;
ωV_d	Direct component of the motor input voltage vector;
ωV_q	In-quadrature component of the motor input voltage vector;
ωI_d	Direct component of the motor output current vector;
ωI_q	In-quadrature component of the motor output current vector;
τ_e	External load torque applied to the motor;

of all the other parameters is reported in Table 1. The last component ωI_{s3} of the current vector ωI_s is proportional to the sum of currents I_{sh} , for $h = \{1, 2, 3\}$, circulating in the three stator phases. In order for the components of the torque vector $\omega K_\tau(\theta)$ to be constant, that is in order for the PMSM to be able to generate a constant torque if the current vector ωI_s is constant, the rotor flux has to be sinusoidal. In this way, only the in-quadrature component ωK_q of the torque vector $\omega K_\tau(\theta)$ is different from zero. Furthermore, under the hypothesis of star connected phases, a static constraint between the stator currents originates $\sum_{h=1}^3 I_{sh} = 0$. This causes the third component ωI_{s3} of the current vector ωI_s to be equal to zero, which in turn implies that the electric part of the PMSM is described by a system having dynamic dimension equal to two. Therefore, under the hypothesis of sinusoidal rotor flux and star-connected stator phases, the state-space model of the PMSM in the d-q frame in (1) can be rewritten as:

$$\underbrace{\begin{bmatrix} pL_{se} & 0 & 0 \\ 0 & pL_{se} & 0 \\ 0 & 0 & J_m \end{bmatrix}}_L \underbrace{\begin{bmatrix} \dot{\omega}_{I_d} \\ \dot{\omega}_{I_q} \\ \dot{\omega}_m \end{bmatrix}}_{\dot{x}} = \underbrace{\begin{bmatrix} pR_s & -p^2\omega_m L_{se} & 0 \\ p^2\omega_m L_{se} & pR_s & \omega K_q \\ 0 & -\omega K_q & b_m \end{bmatrix}}_A \underbrace{\begin{bmatrix} \omega_{I_d} \\ \omega_{I_q} \\ \omega_m \end{bmatrix}}_x + \underbrace{\begin{bmatrix} \omega V_d \\ \omega V_q \\ -\tau_e \end{bmatrix}}_u \quad (2)$$

In steady-state conditions, the PMSM steady-state model (2) becomes [30]:

$$0 = \begin{bmatrix} pR_s & -p^2\omega_m L_{se} & 0 \\ p^2\omega_m L_{se} & pR_s & \omega K_q \\ \textcircled{0} & -\omega K_q & b_m \end{bmatrix} \underbrace{\begin{bmatrix} \omega_{I_d} \\ \omega_{I_q} \\ \omega_m \end{bmatrix}}_{x_{ss}} - \underbrace{\begin{bmatrix} \omega V_d \\ \omega V_q \\ -\tau_e \end{bmatrix}}_u \quad (3)$$

where the parameters meaning is described in Table 1.

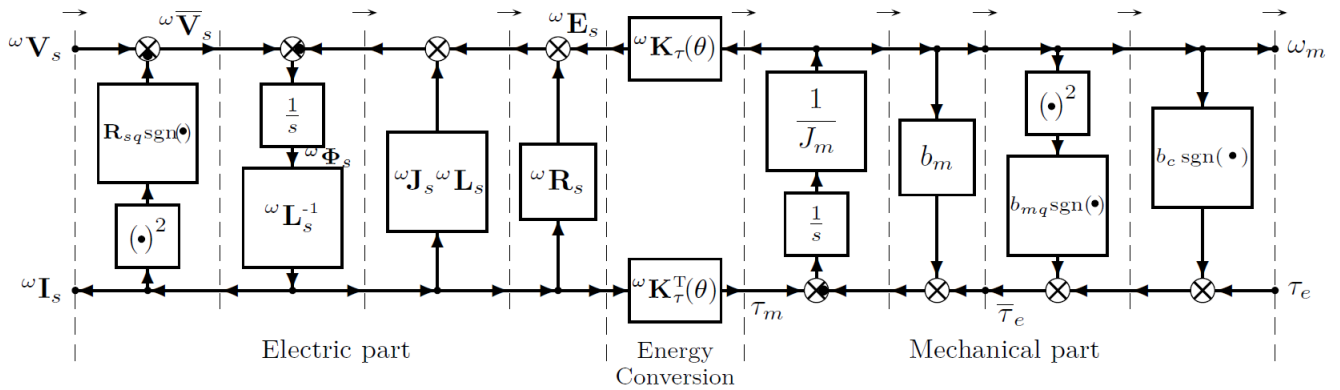


FIGURE 1. POG block scheme of the PMSM in the d-q frame with additional nonlinear friction terms.

Because the term highlighted in red in (3) is zero, it is convenient to enforce $\omega I_d = 0$ since this is the condition maximizing the motor efficiency, as proven in [30]. Under this condition and doing some algebra, system (3) is reduced to:

$$\underbrace{\begin{bmatrix} \omega I_q \\ \omega_m \end{bmatrix}}_y = \underbrace{\begin{bmatrix} \frac{b_m}{p R_s b_m + \omega K_q^2} & \frac{\omega K_q}{p R_s b_m + \omega K_q^2} \\ \frac{\omega K_q}{p R_s b_m + \omega K_q^2} & \frac{1}{p R_s} \end{bmatrix}}_{H_0} \underbrace{\begin{bmatrix} \omega V_q \\ \tau_e \end{bmatrix}}_u \quad (4)$$

If $\omega I_d = 0$ the PMSM behaves, *in first approximation*, as a linear system and its maximum efficiency E^* [30]- [31] is given by [30]-Eq. (30). At steady-state, the PMSM efficiency E when working as a motor, i.e. with a power flow oriented from section $(y_1, u_1) = (\omega I_q, \omega V_q)$ to section $(y_2, u_2) = (\omega_m, \tau_e)$, is:

$$E = \frac{u_2 y_2}{u_1 y_1} = \frac{\tau_e \omega_m}{\omega V_q \omega I_q} \quad (5)$$

In order for the considered model to reproduce an actual PMSM efficiency map, the linear approximation is not sufficient, and some nonlinear dissipative terms have to be added. A graphical representation of the PMSM dynamics including the needed nonlinear dissipative terms is given by the POG block scheme reported in Fig. 1. The power variables ωV_s and τ_e in (4) must be replaced by $\omega \bar{V}_s$ and $\bar{\tau}_e$, defined as:

$$\begin{aligned} \omega \bar{V}_s &= \omega V_s - R_{sq} \operatorname{sgn}(\omega I_s) \omega I_s^2, \\ \bar{\tau}_e &= \tau_e + b_c \operatorname{sgn}(\omega_m) + b_{mq} \operatorname{sgn}(\omega_m) \omega_m^2. \end{aligned} \quad (6)$$

Since $\omega I_d = 0$, the first equation in (6) becomes:

$$\omega \bar{V}_q = \omega V_q - R_{sq} \operatorname{sgn}(\omega I_q) \omega I_q^2. \quad (7)$$

The nonlinear friction terms in (6) and (7) are: the Coulomb friction $b_c \operatorname{sgn}(\omega_m)$, the air friction $b_{mq} \operatorname{sgn}(\omega_m) \omega_m^2$ and the Joule dissipation $R_{sq} \operatorname{sgn}(\omega I_q) \omega I_q^2$. By rewriting (4) as a function of τ_e and ω_m and replacing τ_e and ωV_q with $\bar{\tau}_e$ and

$\omega \bar{V}_q$ in (6) and (7), one obtains:

$$\begin{bmatrix} \omega \bar{V}_q \\ \omega I_q \end{bmatrix} = \begin{bmatrix} \frac{R_s p}{\omega K_q} & \frac{\omega K_q^2 + p R_s b_m}{\omega K_q} \\ \frac{1}{\omega K_q} & \frac{b_m}{\omega K_q} \end{bmatrix} \begin{bmatrix} \bar{\tau}_e \\ \omega_m \end{bmatrix}. \quad (8)$$

III. ESTIMATION ROUTINE APP: INTERFACE

The main user interface of the PMSM parameters estimation Matlab App is shown in Fig. 2, where the reported parameters and results refer to a fictitious machine. The installation file for the proposed Matlab App, created with Matlab R2018b, is named `PMSM_parameters_estimation.mlappinstall` and can be found in the supplementary material in the dataset [17] \Estimation_Matlab_App. The operations the user needs to perform before the estimation begins are summarized in the flowchart of Fig. 3 and detailed in the next subsection.

A. TO BE DEFINED BEFORE THE ESTIMATION BEGINS

Fig. 2 clearly shows the parameters to be estimated in the enlarged magenta window, which can be subdivided into two groups: known and unknown. If a dissipative parameter is known, its selection in the “Estimate?” window must be set to “No” and the corresponding input value will be interpreted as the exact one. If a dissipative parameter is unknown, its selection in the “Estimate?” window must be set to “Yes”, the parameter will be estimated and the given value will be used as initial condition. The torque constant ωK_q is treated as a special case, as it allows one more choice for the user: • if the ωK_q knob is set to “Yes”, ωK_q is unknown and will be *estimated* as in Sec. IV-A; • if “No” is selected, ωK_q is unknown and will be *computed* as in Sec. IV-B; • if “User Defined” is selected, ωK_q is known and the provided value will be interpreted as the exact final one. In the “Define the following parameters:” window, some information has to be user-defined. Among this, note that “Speed at Drop Torque: W_jump ” is the machine speed at which the motor

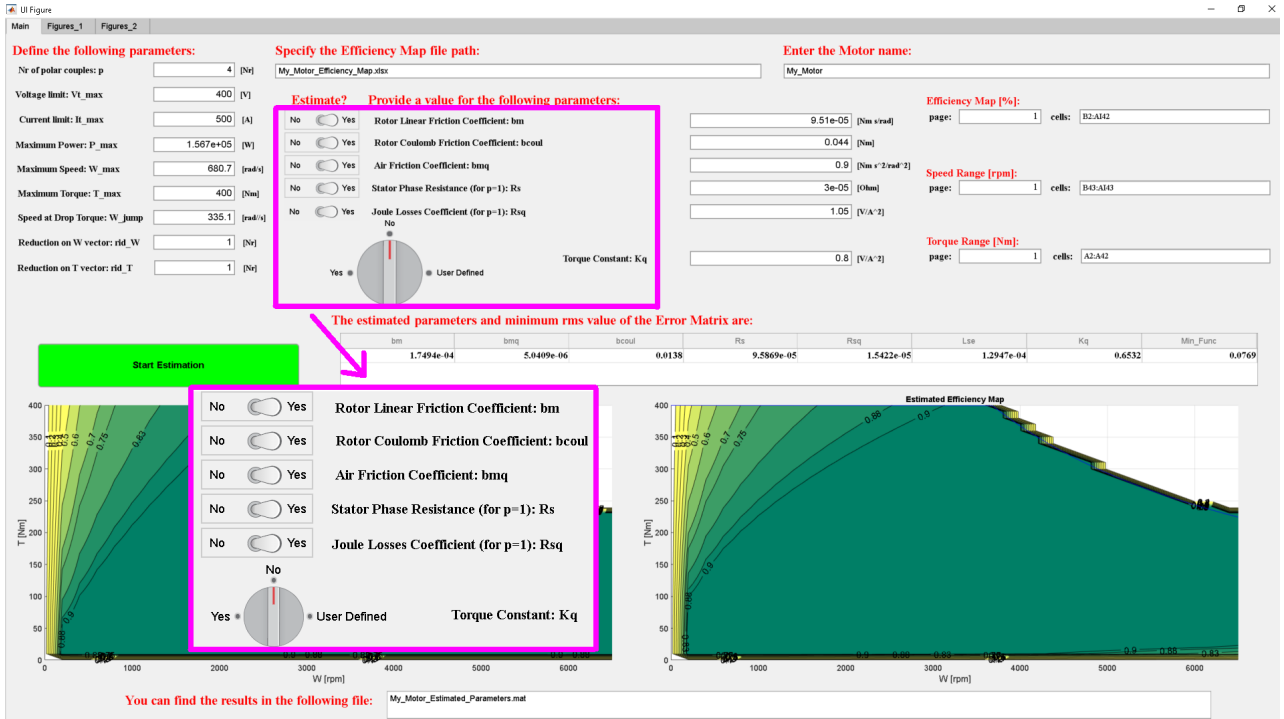


FIGURE 2. Home screen of the user interface created in app designer for the estimation routine.

torque drops following the constant maximum power curve (ω_{mi} in Fig. 4), whereas “Reduction on W vector: rid_W” and “Reduction on T vector: rid_T” are two positive integer numbers, $\in \{1 \dots, m\}$ and $\in \{1 \dots, n\}$ respectively, used to reduce the number of elements in the speed and torque vectors $\omega_m \in \mathbb{R}^{(m \times 1)}$ and $\tau_e \in \mathbb{R}^{(n \times 1)}$ introduced in Sec. IV-A.

B. RESULTS OF THE ESTIMATION

The **Start Estimation** button in Fig. 2 starts the estimation procedure and turns into **Start Estimation** until the estimation ends. The resulting estimated parameters are shown in the “The estimated parameters and minimum rms value of the Error Matrix are:” window, together with the final value “Min_Func” of the objective function Q in (11). The left Matlab figure shows the actual efficiency map $E_{map_{act}}(\omega_m, \tau_e)$ provided by the input Excel file. The right output Matlab figure shows the estimated efficiency map $E_{map_{est}}(\omega_m, \tau_e)$ computed as described in Sec. IV-A and the maximum torque vs speed characteristic $\tau_{e_{max}}(\omega_m)$ computed as described in Sec. IV-E. Other interesting output figures are provided by the Matlab App in the “Figures_1” and “Figures_2” screens, see PMSM_parameters_estimation.mlappinstall in [17]Estimation_Matlab_App, such as the distribution of vectors ${}^\omega V_s$ and ${}^\omega I_s$ vs (ω_m, τ_e) , the $E_{map_{error}}$ on plane (ω_m, τ_e) defined in (10), the resistive voltage due to R_s and to R_s and R_{sq} together, and the resistive torque due to b_m and to b_m, b_{mq} and b_{coul} together.

IV. ESTIMATION ROUTINE: THEORY

In this section, the theory and the implementation of the PMSM parameters estimation routine are analyzed and commented.

A. ESTIMATION OF DISSIPATIVE TERMS AND ${}^\omega K_q$

Out of $b_m, R_s, b_c, b_{mq}, R_{sq}$ and ${}^\omega K_q$, the user can select the unknown ones to be estimated, see Sec. III-A. The estimation makes use of the actual efficiency map $E_{map_{act}}$ of the considered PMSM. The actual efficiency map $E_{map_{act}}$ is, in general, a rectangular matrix $E_{map_{act}} \in \mathbb{R}^{(n \times m)}$, where n is the number of elements in the torque vector τ_e and m is the number of elements in the speed vector ω_m ; $\tau_e \in \mathbb{R}^{(n \times 1)}, \omega_m \in \mathbb{R}^{(m \times 1)}$. Using the user defined values of the parameters, representing the *initial value* for the unknown ones and the *exact final value* for the known ones, the routine loops over the set of operating points in the output plane $(y_2, u_2) = (\omega_m, \tau_e) \in S$, where the set S is defined as:

$$S = \{(\omega_{mi}, \tau_{ej}) \text{ s.t. } \omega_{mi} \tau_{ej} \leq P_{max}\},$$

$$\text{for } i \in \{1, 2, \dots, m\} \text{ } j \in \{1, 2, \dots, n\}, \quad (9)$$

with P_{max} being the maximum motor power. For simplicity of notation, let S_h denote the generic couple (ω_{mi}, τ_{ej}) , for i and j in (9). The routine exploits (6) and (8) to compute the couple $(y_1, u_1) = ({}^\omega I_{qs_h}, {}^\omega V_{qs_h})$, associated with the considered operating point on the output plane $S_h \in S$. By using (5), the estimated efficiency $E_{map_{est_{S_h}}}$ of the PMSM in $S_h \in S$ can be computed. By looping over all the output plane points $S_h \in S$, the estimation routine builds the estimated efficiency map

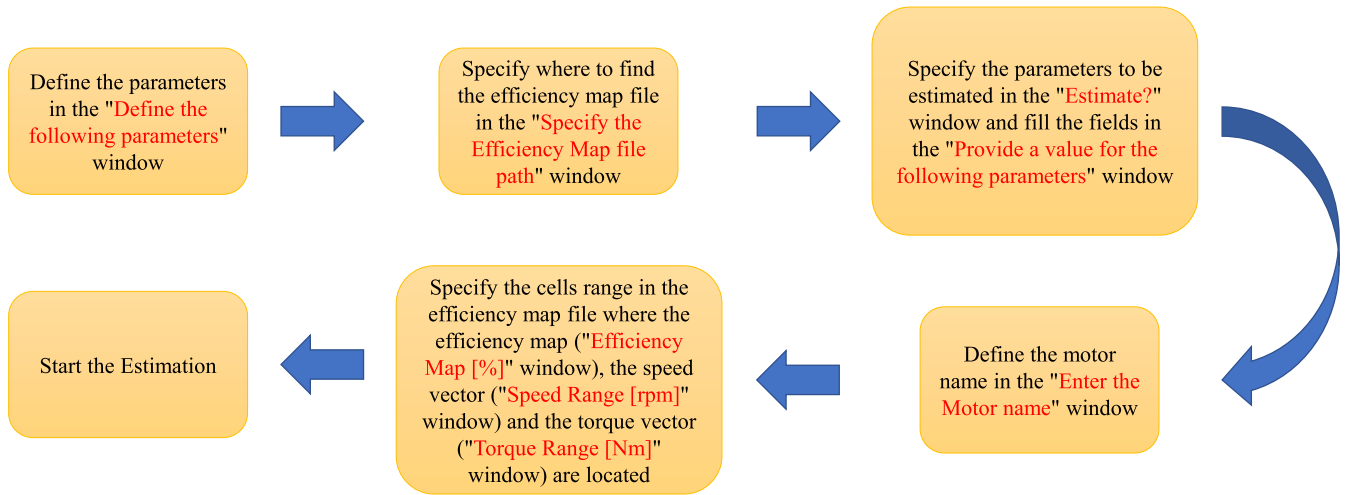


FIGURE 3. Operations to be performed before the estimation begins.

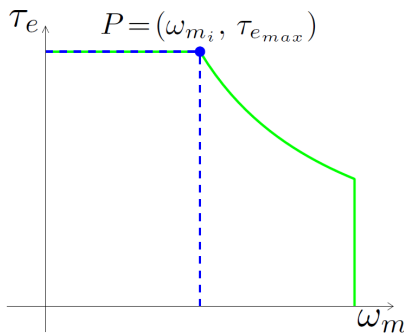


FIGURE 4. Point $P = (\omega_{m_i}, \tau_{e_{max}})$ on the maximum torque vs speed characteristic $\tau_{e_{max}}(\omega_m)$.

$E_{map_{est}}$. The remaining empty elements $E_{map_{est}S_h}$, for $S_h \notin S$, are set to zero. The efficiency error map is then computed:

$$E_{map_{error}} = |E_{map_{act}} - E_{map_{est}}|. \quad (10)$$

The `fminsearch` Matlab function is then employed in order to perform a defined number of iterations allowing to identify the values of the parameters that the user has selected to estimate, in order to minimize the function:

$$Q = \sqrt{\left(\frac{1}{r}\right) \sum_{h=1}^r (E_{map_{error}S_h})^2}, \quad r = \dim(S). \quad (11)$$

B. COMPUTATION OF THE TORQUE CONSTANT ${}^\omega K_q$

In the “ ${}^\omega K_q$ knob \rightarrow No” special case in Sec. III-A, ${}^\omega K_q$ is computed as follows. Reference is made to the case of maximum torque generation $\tau_{e_i} = \tau_{e_{max}}$, i.e. $|{}^\omega I_s| = |{}^\omega I_{s_{max}}|$. Assuming ${}^\omega I_d = 0$ and star-connected phases, $|{}^\omega I_s| = |{}^\omega I_q|$ holds. From (3) and $\tau_m = {}^\omega K_q {}^\omega I_q$ (see Fig. 1), and neglecting the dissipative terms affecting the mechanical part b_m, b_c and b_{mq} (for the computation of ${}^\omega K_q$ only) introduced in Fig. 1, (6) and (7), it follows ${}^\omega K_q = \tau_{e_{max}}/|{}^\omega I_q| = \tau_{e_{max}}/|{}^\omega I_{s_{max}}|$.

C. COMPUTATION OF THE DYNAMIC TERM L_{se}

Let us assume $\tau_{e_j} = \tau_{e_{max}}$, for which $|{}^\omega I_s| = |{}^\omega I_{s_{max}}|$ holds. Let operating point P on the output plane:

$$P = (y_2, u_2) = (\omega_{m_i}, \tau_{e_{max}}) \quad (12)$$

be the one giving the maximum motor torque and power: $\tau_e = \tau_{e_{max}}$ and $P_m = P_{max}$. Point P is shown on the maximum torque vs speed characteristic $\tau_{e_{max}}(\omega_m)$ of the motor reported in Fig. 4. Assuming ${}^\omega I_d = 0$ and star-connected phases, $|{}^\omega I_s| = |{}^\omega I_q|$ holds. From (3), $\tau_m = {}^\omega K_q {}^\omega I_q$, (6), (7), (8), and using point P in (12), it is:

$$\begin{aligned} |{}^\omega I_{q_{max}}| &= |{}^\omega I_{s_{max}}| \\ &= \frac{\tau_e + b_c \operatorname{sgn}(\omega_{m_j}) + b_{mq} \operatorname{sgn}(\omega_{m_j}) \omega_{m_j}^2 + b_m \omega_{m_j}}{\omega K_q}. \end{aligned}$$

From (3), (6), (7) and (8), one can notice that:

$${}^\omega V_{q_j} = p R_s {}^\omega I_{q_{max}} + {}^\omega K_q \omega_{m_j} + R_{sq} \operatorname{sgn}({}^\omega I_{q_{max}}) {}^\omega I_{q_{max}}^2.$$

By assuming $|{}^\omega V_s| = |{}^\omega V_{s_{max}}|$ in the considered operating point $(\omega_{m_j}, \tau_{e_{max}}) \in S$, and assuming star-connected stator phases, one obtains: ${}^\omega V_{d_j} = \sqrt{|{}^\omega V_{s_{max}}|^2 - {}^\omega V_{d_j}^2}$. From the first equation in (3), it follows:

$$L_{se} = \left| \frac{{}^\omega V_{d_j}}{-p^2 \omega_m {}^\omega I_{q_j}} \right|. \quad (13)$$

From (13), parameter L_{se} is computed.

D. DISTRIBUTION OF $|{}^\omega V_s|$ AND $|{}^\omega I_s|$ ON (ω_m, τ_e)

After the estimation, Eqs. (6), (7) and (8) are once again exploited to compute the couple $({}^\omega I_{q_{S_h}}, {}^\omega V_{q_{S_h}})$ associated with all the operating points on the output plane $S_h \in S$ in (9). Since ${}^\omega I_d = 0$ is enforced, $|{}^\omega I_s|(\omega_m, \tau_e) = |{}^\omega I_q|(\omega_m, \tau_e)$ holds. The latter is provided as output in the top-right figure of the “Figures_1” results screen. As for the voltage vector ${}^\omega V_s$,

FIGURE 5. PMSM superblock user interface.

the remaining direct component ${}^{\omega}V_d$ needs to be computed by using the first equation of (3). From ${}^{\omega}V_d$ and ${}^{\omega}V_q$, the distribution $|{}^{\omega}V_s|(\omega_m, \tau_e)$ can be computed. The latter is provided as output in the top-left figures of the “Figures_1” and “Figures_2” results screens.

E. Derivation of characteristic $\tau_{e_{max}}(\omega_m)$

From $\tau_m = {}^{\omega}K_q {}^{\omega}I_q$ and ${}^{\omega}I_d = 0$, it follows that the maximum motive torque is achieved when $|{}^{\omega}I_q| = |{}^{\omega}I_{s_{max}}|$. Additionally, from the third equation of (3) and from the second equation of (6) it is known that, at steady-state, the motive torque $\tau_m = {}^{\omega}K_q {}^{\omega}I_q$ and the external torque τ_e are related to each other through b_m , b_c and b_{mq} . Since ${}^{\omega}I_d = 0$, let us define a new in-quadrature current component vector $|{}^{\omega}I_{q_{vect}}|$ covering the whole range from 0 to $|{}^{\omega}I_{s_{max}}|$ with steps of 1A, and let us call i the index looping over this vector covering the whole range of torque τ_e from zero up to $\tau_{e_{max}}$ thanks to the relation between the motive torque τ_m and the external torque τ_e highlighted above. The computation of the different torque versus speed characteristics $\tau_e(\omega_m)$ up to $\tau_{e_{max}}(\omega_m)$ for all current values within $|{}^{\omega}I_{q_{vect}}|$ can then be performed.

V. PMSM SUPERBLOCK

This section describes the mask interface of the PMSM superblock reported in Fig. 5, which is subdivided into two containers: “Machine Parameters” and “Tunable Popups”. The PMSM superblock is provided in the Simulink block schemes `EM_Speed_Control_slx.slx` and `EM_Torque_Control_slx.slx` in [17]\PMSM_Superblock created with Matlab R2018B. The “Machine Parameters” container minimal configuration is shown in Fig. 5, reporting the parameters that always have to be defined, whereas more parameters appear selecting the different options in the “Tunable Popups” container. The parameters meaning is described in the “Help” section

of the PMSM superblock interface. The parameters to be defined in the “Tunable Popups” container are: Star-Connected Phases? Yes or No \Rightarrow {Yes, No}, Torque Limit? \Rightarrow {Tq_Limit_On, Tq_Limit_Off}, Speed Control/Torque Control? \Rightarrow {Speed_Control, Torque_Control}, ThreePhase/ dq_Frame? \Rightarrow {ThreePhase, dq_Frame}, Mechanical Dynamics: Yes or No? \Rightarrow {Mech_Dynamics_Yes, Mech_Dynamics_No}. The names of the popups reflect their meaning, and more details are also given in the “Help” section of the PMSM superblock interface, together with the new parameters to be defined in the “Machine Parameters” container when selecting the different “Tunable Popups” options. The tunable popup “Star-Connected Phases? Yes or No” must be set to “Yes” in order to interface the PMSM superblock with the parameters estimation routine. The external view of the PMSM superblock also changes depending on the options selected for the tunable popups. For the sake of brevity, only the external view of the PMSM superblock with one popups configuration is shown in this paper, see Fig. 6. The Reader is invited to try the other options in the Simulink block schemes `EM_Speed_Control_slx.slx` and `EM_Torque_Control_slx.slx` in [17]\PMSM_Superblock.

When compared with other simulation tools, such as the models made available by the Matlab/Simulink [12]- [13] for example, the proposed PMSM superblock offers some advantages. As far as the mechanical part of the motor is concerned, the proposed PMSM superblock accounts for one more friction effect, namely the air friction, exhibiting a square dependence on the motor speed. As far as the electrical part of the motor is concerned, the proposed PMSM superblock accounts for one more dissipative effect, namely the Joule losses, exhibiting a square dependence on the current vector components. These two dissipative effects are important when reproducing the efficiency map

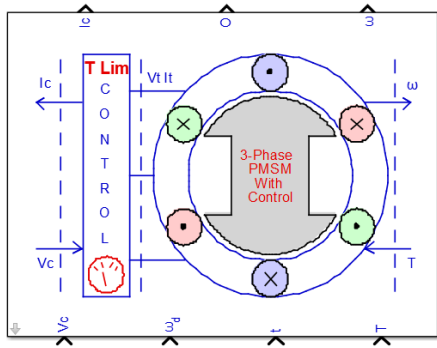


FIGURE 6. PMSM superblock external view with “Tq.Limit_On”, “Speed_Control”, “ThreePhase” and “Mech_Dynamics_Yes” selected.

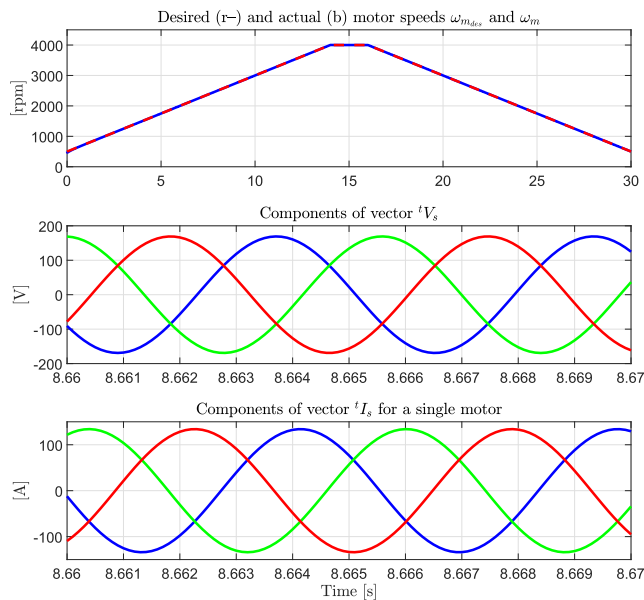


FIGURE 7. Simulation performed in EM_Speed_Control.m.

of an actual PMSM in the different operating points. Furthermore, an important advantage of the proposed PMSM superblock is the fact that it can be directly used in conjunction with the proposed PMSM parameters estimation Matlab App [17]Estimation_Matlab_App. The user can first use the PMSM parameters estimation Matlab App to estimate the motor parameters from the efficiency map and then directly feed the estimated parameters to the PMSM superblock.

A. SIMULATIONS

The supplementary material in the dataset [17]PMSM_Superblock reports two scripts EM_Speed_Control.m and EM_Torque_Control.m, simulating the Simulink models EM_Speed_Control_slx.slx and EM_Torque_Control_slx.slx, respectively. The two scripts contain two simulation examples applying a speed and a torque control to the PMSM, respectively. The PMSM parameters employed in this case are those of the fictitious machine

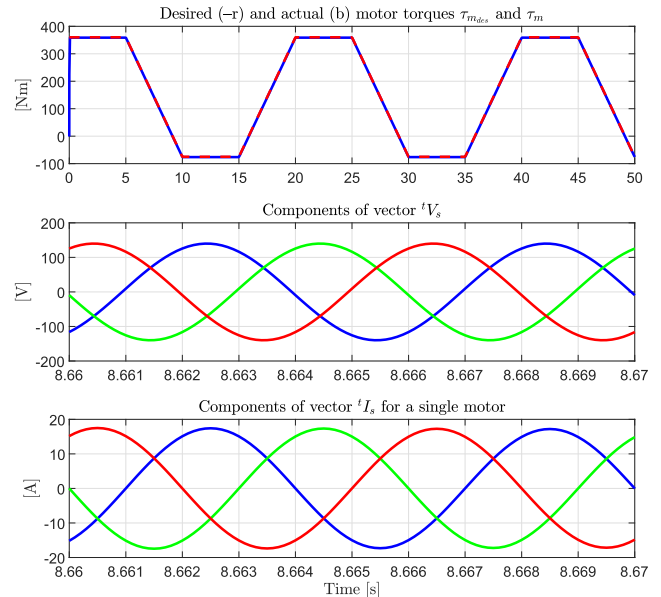


FIGURE 8. Simulation performed in EM_Torque_Control.m.

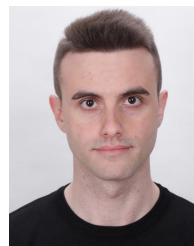
reported in My_Motor_Estimated_Parameters.mat, estimated in [17]Estimation_Matlab_App. Both in EM_Speed_Control.m and in EM_Torque_Control.m, the tunable popup options chosen in the two cases are set from line 10 to line 15 of the scripts, but can also be manually set from the PMSM superblock user interface itself. The remaining simulation parameters are set in the remaining part of the two scripts. Note that the tunable popup options set in EM_Speed_Control.m are such that the PMSM superblock external view in EM_Speed_Control_slx.slx is exactly the one illustrated in Fig. 6. Some results for the two simulation case studies in EM_Speed_Control.m and EM_Torque_Control.m are reported in Fig. 7 and Fig. 8. The upper subplot of Fig. 7 shows the desired and actual motor speeds $\omega_{m_{des}}$ and ω_m in red dashed and blue lines, respectively. The middle subplot of Fig. 7 shows the three components of the voltage vector tV_s in the original three-phase frame. The lower subplot of Fig. 7 shows the three components of the current vector tI_s in the original three-phase frame for a single motor, as three PMSMs are working in parallel in the considered simulation examples, see parameter Nr_mot in EM_Speed_Control.m and EM_Torque_Control.m. The upper subplot of Fig. 8 shows the desired and actual motors torques $\tau_{m_{des}}$ and τ_m (the latter also accounting for the mechanical friction contributions) in red dashed and blue lines, respectively. The middle and lower subplots of Fig. 8 report the same information as those of Fig. 7 but for the torque control case. From the upper subplots of Fig. 7 and Fig. 8, it is possible to see the good superposition between the desired and actual characteristics, highlighting the effectiveness of the control.

VI. CONCLUSION

In this paper, a Matlab App created for PMSM parameters estimation has been presented. The App allows to estimate electrical and mechanical parameters and can be potentially implemented online. Next, a generalized PMSM Simulink superblock with a user-friendly interface has been introduced. The PMSM superblock is compatible with the parameters estimation Matlab App, allows both speed and torque control, and lets the user choose whether to compute the three-phase currents and voltages and if the rotor mechanical dynamics has to be included in the simulation or not. The results of two simulations performed on the PMSM superblock are finally reported, applying a speed and a torque control, respectively.

REFERENCES

- [1] Z. Ping, T. Wang, Y. Huang, H. Wang, J.-G. Lu, and Y. Li, "Internal model control of PMSM position servo system: Theory and experimental results," *IEEE Trans. Ind. Informat.*, vol. 16, no. 4, pp. 2202–2211, Apr. 2020.
- [2] S. Murshid and B. Singh, "A multiobjective GI-based control for effective operation of PV pumping system under abnormal grid conditions," *IEEE Trans. Ind. Informat.*, vol. 16, no. 11, pp. 6880–6891, Nov. 2020.
- [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 Trans. Ind. Electron.*, vol. 59, no. 2, pp. 803–811, Feb. 2012.
- [4] Y. Dai, L. Song, and S. Cui, "Development of PMSM drives for hybrid electric car applications," *IEEE Trans. Magn.*, vol. 43, no. 1, pp. 434–437, Jan. 2007.
- [5] M. S. Rafaq and J.-W. Jung, "A comprehensive review of state-of-the-art parameter estimation techniques for permanent magnet synchronous motors in wide speed range," *IEEE Trans. Ind. Informat.*, vol. 16, no. 7, pp. 4747–4758, Jul. 2020.
- [6] M. Tian, B. Wang, Y. Yu, Q. Dong, and D. Xu, "Discrete-time repetitive control-based ADRC for current loop disturbances suppression of PMSM drives," *IEEE Trans. Ind. Informat.*, vol. 18, no. 5, pp. 3138–3149, May 2022.
- [7] S.-K. Kim and C. K. Ahn, "Active-damping speed tracking technique for permanent magnet synchronous motors with transient performance boosting mechanism," *IEEE Trans. Ind. Informat.*, vol. 18, no. 4, pp. 2171–2179, Apr. 2022.
- [8] A. Brosch, O. Wallscheid, and J. Böcker, "Torque and inductances estimation for finite model predictive control of highly utilized permanent magnet synchronous motors," *IEEE Trans. Ind. Informat.*, vol. 17, no. 12, pp. 8080–8091, Dec. 2021.
- [9] A. Favato, P. G. Carlet, F. Toso, R. Torchio, L. Ortombina, M. Bruschetta, R. Carli, P. Alotto, S. Bolognani, and J. Rodriguez, "Fast solver for implicit continuous set model predictive control of electric drives," *IEEE Access*, vol. 10, pp. 17430–17440, 2022.
- [10] H. Yang, J. Yang, and X. Zhang, "DC-bus capacitor maximum power discharge strategy for EV-PMSM drive system with small safe current," *IEEE Access*, vol. 9, pp. 132158–132167, 2021.
- [11] J. Zhang, Y. Wang, K. Zhu, Y. Zhang, and Y. Li, "Diagnosis of interturn short-circuit faults in permanent magnet synchronous motors based on few-shot learning under a federated learning framework," *IEEE Trans. Ind. Informat.*, vol. 17, no. 12, pp. 8495–8504, Dec. 2021.
- [12] *Simulink Documentation*. [Online]. Available: <https://it.mathworks.com/help/autoblks/ref/interiorpmsm.html>
- [13] *Simulink Documentation*. [Online]. Available: <https://it.mathworks.com/help/autoblks/ref/surfacemountpmsm.html>
- [14] R. Zanasi, "The power-oriented graphs technique: System modeling and basic properties," in *Proc. IEEE Vehicle Power Propuls. Conf. (VPPC)*, Lille, France, Sep. 2010, pp. 1–6.
- [15] X. Roboam, *Systemic Design Methodologies for Electrical Energy Systems: Analysis, Synthesis and Management*. Hoboken, NJ, USA: Wiley, Dec. 2012.
- [16] R. Zanasi, G. H. Geitner, A. Bouscayrol, and W. Lhomme, "Different energetic techniques for modelling traction drives," in *Proc. 9th Int. Conf. Modeling Simulation Electr. Mach., Converters Syst.*, Quebec City, QC, Canada, Jun. 2008, pp. 8–11.
- [17] *Link of the Supplementary Material*. [Online]. Available: http://www.dii.unimo.it/~zanasi/PMSM/Suppl_M.zip. 10.21227/yn64-6c31.
- [18] M. Rashed, P. F. A. MacConnell, A. F. Stronach, and P. Acarnley, "Sensorless indirect-rotor-field-orientation speed control of a permanent-magnet synchronous motor with stator-resistance estimation," *IEEE Trans. Ind. Electron.*, vol. 54, no. 3, pp. 1664–1675, Jun. 2007.
- [19] W. Yu, H. Liang, X. Dong, and Y. Luo, "Modeling and identification of permanent magnet synchronous motor via deterministic learning," *IEEE Access*, vol. 8, pp. 168516–168525, 2020.
- [20] K. Liu, J. Feng, S. Guo, L. Xiao, and Z.-Q. Zhu, "Identification of flux linkage map of permanent magnet synchronous machines under uncertain circuit resistance and inverter nonlinearity," *IEEE Trans. Ind. Informat.*, vol. 14, no. 2, pp. 556–568, Feb. 2018.
- [21] H.-C. Yeh and S.-M. Yang, "Phase inductance and rotor position estimation for sensorless permanent magnet synchronous machine drives at standstill," *IEEE Access*, vol. 9, pp. 32897–32907, 2021.
- [22] K. Choi, Y. Kim, K.-S. Kim, and S.-K. Kim, "Using the stator current ripple model for real-time estimation of full parameters of a permanent magnet synchronous motor," *IEEE Access*, vol. 7, pp. 33369–33379, 2019.
- [23] J. Lee and J.-I. Ha, "Temperature estimation of PMSM using a difference-estimating feedforward neural network," *IEEE Access*, vol. 8, pp. 130855–130865, 2020.
- [24] C. A. Agustin, J.-T. Yu, Y.-S. Cheng, C.-K. Lin, and Y.-W. Yi, "A synchronized current difference updating technique for model-free predictive current control of PMSM drives," *IEEE Access*, vol. 9, pp. 63306–63318, 2021.
- [25] R. Fu, "Robust model predictive flux control of PMSM drive using a compensated stator flux predictor," *IEEE Access*, vol. 9, pp. 136736–136743, 2021.
- [26] M. Burth, G. C. Verghese, and M. Vélez-Reyes, "Subset selection for improved parameter estimation in on-line identification of a synchronous generator," *IEEE Trans. Power Syst.*, vol. 14, no. 1, pp. 218–225, Feb. 1999.
- [27] G. Feng, C. Lai, K. Mukherjee, and N. C. Kar, "Current injection-based online parameter and VSI nonlinearity estimation for PMSM drives using current and voltage DC components," *IEEE Trans. Transport. Electrific.*, vol. 2, no. 2, pp. 119–128, Jun. 2016.
- [28] G. Feng, C. Lai, and N. C. Kar, "A novel current injection-based online parameter estimation method for PMSMs considering magnetic saturation," *IEEE Trans. Magn.*, vol. 52, no. 7, Jul. 2016, Art. no. 8106004.
- [29] X. Chen, J. Wang, and A. Griffo, "A high-fidelity and computationally efficient electrothermally coupled model for interior permanent-magnet machines in electric vehicle traction applications," *IEEE Trans. Transport. Electrific.*, vol. 1, no. 4, pp. 336–347, Dec. 2015.
- [30] R. Zanasi and D. Tebaldi, "Power flow efficiency of linear and nonlinear physical systems," in *Proc. 18th Eur. Control Conf. (ECC)*, Naples, Italy, Jun. 2019, pp. 3829–3835.
- [31] R. Zanasi and D. Tebaldi, "Study of the bidirectional efficiency of linear and nonlinear physical systems," in *Proc. 45th Annu. Conf. IEEE Ind. Electron. Soc. (IECON)*, Lisbon, Portugal, Oct. 2019, pp. 545–552.



DAVIDE TEBALDI was born in Pavullo nel Frignano, Italy, in 1992. He received the B.S. and M.S. degrees (*cum laude*) in electronic engineering and the Ph.D. degree in systems and control engineering from the University of Modena and Reggio Emilia, Italy, in 2015, 2018, and 2022, respectively. He is currently a Postdoctoral Researcher at the University of Modena and Reggio Emilia. His research interests include energetic modeling, control and simulation of mechatronic systems, with main application in automotive, electrical machines, and power electronics fields.