

Multiagent Control in Modular Motor Drives by Means of Deterministic Consensus

Lynn Verkroost¹, Frederik De Belie¹, Peter Sergeant¹, *Senior Member, IEEE*,
and Hendrik Vansompel¹

Abstract—A modular motor drive (MMD) is composed of multiple identical pole drive units (PDUs), which consist of a concentrated stator winding fed by a dedicated power electronic converter module. To exploit the full flexibility and reliability of these MMDs, they are equipped with multiple controllers, operating as synchronized peers, and each driving only a limited set of PDUs. An MMD can hence be considered as a multiagent system, in which all the agents cooperate to deliver a certain torque demand. The goal of this work is to distribute this torque demand among the agents of a modular drive in a completely decentralized way, in order to comply with the flexibility and reliability of the hardware, while still taking into account the present condition of each agent. A deterministic consensus algorithm is used for this purpose, in combination with decentralized proportional integral (PI) current controllers. Communication is only allowed between neighboring agents, to limit the communication load. It is shown in this article that the total torque demand is delivered, under healthy conditions as well as during agent malfunctions and with non-identical agents. The concept is demonstrated on a modular axial-flux permanent magnet synchronous machine with 15 PDUs that are assigned to five agents.

Index Terms—Consensus, distributed control, fault tolerance, modular motor drive (MMD), multiagent system.

Manuscript received 6 September 2021; revised 2 December 2021, 2 February 2022, and 24 February 2022; accepted 21 March 2022. Date of publication 5 April 2022; date of current version 5 October 2022. This work was supported by the Research Foundation-Flanders (FWO). L. Verkroost was awarded a Ph.D. fellowship strategic basic research from the FWO in 2019 under Grant 3S045319. This work is part of the ModulAr SBO project funded and supported by Flanders Make vzw, the strategic research centre for the manufacturing industry. (Corresponding author: Lynn Verkroost.)

Lynn Verkroost is with the Electrical Energy Laboratory, Department of Electromechanical, Systems and Metal Engineering, Ghent University, 9000 Ghent, Belgium, with Flanders Make@UGent-EEDT-MP, 3001 Leuven, Belgium, and also with Flanders Make@UGent-EEDT-DC, 3001 Leuven, Belgium (e-mail: lynn.verkroost@ugent.be).

Frederik De Belie is with the Electrical Energy Laboratory, Department of Electromechanical, Systems and Metal Engineering, Ghent University, 9000 Ghent, Belgium, and also with Flanders Make@UGent-EEDT-DC, 3001 Leuven, Belgium (e-mail: frederik.debelie@ugent.be).

Peter Sergeant and Hendrik Vansompel are with the Electrical Energy Laboratory, Department of Electromechanical, Systems and Metal Engineering, Ghent University, 9000 Ghent, Belgium, and also with Flanders Make@UGent-EEDT-MP, 3001 Leuven, Belgium (e-mail: peter.sergeant@ugent.be; hendrik.vansompel@ugent.be).

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TIE.2022.3163504>.

Digital Object Identifier 10.1109/TIE.2022.3163504

I. INTRODUCTION

MODULAR motor drives (MMDs) are created by segmenting an electric motor and its power electronic (PE) converter into n identical modules. These modules, which are often called pole drive units (PDUs) in literature, consist of a stator core element and a concentrated winding, fed by a dedicated PE converter [1], [2]. The PDUs can be seen as building blocks for electric drives, offering flexibility and reliability, and an advantageous economies of scale effect. Beneficial effects of this segmentation include fault tolerance, additional degrees-of-freedom for advanced control strategies, increased efficiency, power rating scalability, better thermal performance, and a possible reduction in the overall size and cost of the PE (due to the possibility to eliminate the inverter heat sink [3], the reduction in voltage and/or current rating of the power semiconductor devices [2], [4], and the possibility to reduce the dc link capacitor size [3], [5]) in comparison with traditional drives [2]–[7].

To exploit the full range of possibilities of these modular drives, not only the motor and its PE should be modular, but also the control. In literature, MMDs are often controlled by means of one central controller based on vector space decomposition [8]–[10]. Although these centralized control strategies result in excellent performance of the drive, they do introduce a single point of failure in the drive (i.e., the centralized controller itself). Besides, a centralized controller also needs to be redesigned completely if one of the PDUs is changed or faulted. To overcome these issues, MMDs can be equipped with multiple controllers as well, which operate as synchronized peers to run identical control programs [1], [3]. Under this distributed control strategy, each of the p controllers drives only a limited set of m PDUs ($n = m \cdot p$). A choice that is frequently made in literature, is to equip each set of three PDUs with an independent three-phase controller (i.e., $m = 3$) [11], [12]. In this way, the MMD can benefit from the well-consolidated three-phase technology and standard three-phase control strategies, as well as from the fault tolerance, flexibility, and redundancy of modularity [13].

A very useful property of these modular multithree-phase machines, is that each set of PDUs can contribute differently to a total torque demand, which is equivalent to a certain power demand under constant speed. The total torque demand can be converted into specific current set-points, according to so-called sharing coefficients [6], [8]–[10], [14], [15]. Different sharing coefficients result in different current amplitudes in the different

sets of PDUs. This feature proves for instance very useful in case of a fault in one of the motor- or PE modules: the current in the corresponding set is set equal to zero, while the healthy sets increase their current amplitudes—within their current limits—to satisfy the torque or power demand [16]. However, to the authors' best knowledge, the literature still lacks a method to determine these sharing coefficients online, without the need for a master or a central supervisory controller, and with the possibility for each set of PDUs to adapt its sharing coefficient based on its own knowledge of its present condition.

The combination of a set of PDUs and its controller in an MMD can be seen as an agent in a multiagent system. In such a network, the autonomous agents can cooperate to perform distributed optimization [17]–[20], to reject disturbances [21], [22], to compensate for failures [23], or, in general, to reach agreement on a specific quantity of interest [24]. The challenge in which the agents seek to agree on a linear combination of static parameters, is commonly referred to as the static average consensus problem [25], [26]. The multiagent system is equipped with a communication network for this purpose. The local information exchange enables the agents to compute the average of a set of fixed quantities, based on local computations and partial information only. In literature, a large variety of methods can be found for solving the static average consensus problem, each with their own requirements on the agents, the network topology, and the timing and nature of the communication [27], [28].

The goal of this work is hence to use a deterministic static average consensus algorithm to distribute a torque demand among the agents of an MMD in a completely decentralized way. The consensus algorithm that is used in this work, is inspired on the consensus based approach for the economic dispatch problem in a smart grid, which determines the power outputs of all the generators in the grid in a distributed way [29]–[31]. It is supposed that all the agents in the MMD exchange data with their two closest neighbors within one update cycle of the algorithm, after which they perform local calculations to reach agreement on their incremental costs. This incremental cost is a parameter that indicates for each agent individually how much effort it would take for that agent to further increase its current set-points. The major reasons for the use of this specific consensus algorithm are that the agents are able to decide among themselves which agent delivers which part of a total torque demand, taking into account the current state of each agent, and without the need to introduce a master or a centralized controller. The deterministic consensus algorithm provides each agent with individual current set-points, based on this incremental cost. The m -phase current controller of each agent then controls the currents through its PDUs toward these set-points in a completely autonomous way [11]. Although predictive current controllers are known for their superior dynamic performance, it is opted in this research to make use of decentralized proportional integral (PI) current controllers. These PI current controllers have the advantage that they are less dependent on an accurate machine model, less complex, and hence less computationally expensive than predictive current controllers [32].

Although the methods proposed in this work are applicable for any number of PDUs n and any number of agents p , they will be demonstrated for the specific case of a 4 kW segmented armature

torus (SAT) axial-flux permanent magnet synchronous machine (AFPMSM) with $n = 15$ PDUs, which are grouped into $p = 5$ agents of $m = 3$ PDUs.

The rest of the article is organized as follows. The definition of an MMD architecture is summarized and presented specifically for this case study in Section II. Section III elaborates on the deterministic static average consensus algorithm used to determine the current set-points for the agents, after which the state-of-the-art decentralized PI current control strategy is discussed in Section IV. The proposed control strategy is experimentally validated in Section V, under healthy operation conditions, as well as under agent malfunctions, and under differences between the agents. The effect of the communication frequency on the convergence and settling time of the multiagent current control strategy is discussed in Section VI. Finally, Section VII concludes this article.

II. MODULAR MOTOR DRIVE ARCHITECTURE

A schematic overview of the MMD architecture used throughout this work is presented in Fig. 1. A motor module consists of a stator core element with a concentrated winding. Each of the n motor modules is fed by a dedicated PE module. The assembly of a motor module and a PE module is called a PDU. In this work, a PE module comprises a half-bridge inverter leg and a phase current measurement. The PE modules are connected in parallel to a dc power supply.

The n PDUs are grouped into p sets of m PDUs. Each of these sets is equipped with a standard m -phase (micro)controller. The combination of a set of m PDUs and its controller is called an agent. An agent can hence be seen as a functional entity that can operate independently of the other agents in the system. In this work, the MMD consists of 15 PDUs, which are grouped into five agents of three PDUs. The three PDUs of one agent are connected in one neutral point, as can be seen in Fig. 2. The PDUs are arranged in such a way that they have a phase shift of $2\pi/m = 2\pi/3$ electrical radians within an agent (e.g., in Fig. 2, PDU a has a phase angle ϕ_a of 0 rad, for PDU f $\phi_f = -2\pi/3$ rad, and for PDU k $\phi_k = -4\pi/3$ rad). The PDUs of neighboring agents are shifted with an angle of $2\pi/n = 2\pi/15$ electrical radians (e.g., PDU b has a phase angle $\phi_b = -2\pi/15$ rad, PDU c has a phase angle $\phi_c = -4\pi/15$ rad,...)

Finally, the complete MMD is composed of multiple agents. A communication network is established between the agents. Data exchange is only allowed with the two closest neighbors of each agent, i.e., agent $x \in \{1, \dots, p\}$ can only communicate with agents $x - 1$ and $x + 1$, as is illustrated in Fig. 3(a). This communication network is used to distribute a total current set-point I_q^* (originating from the user, and representing a certain torque demand as explained in Section III) over the p agents by means of a deterministic static average consensus algorithm. In this work, only agents 1 and 3 receive the total current set-point I_q^* from the user. This total set-point I_q^* is then distributed among the five agents by means of the consensus algorithm explained in Section III, resulting in five current set-points $i_{q,x}^*$ ($x \in \{1, \dots, 5\}$). The agent-specific current set-point $i_{q,x}^*$ is then fed to the conventional three-phase field-oriented controller of agent x , as is illustrated and explained in Fig. 2. The decentralized

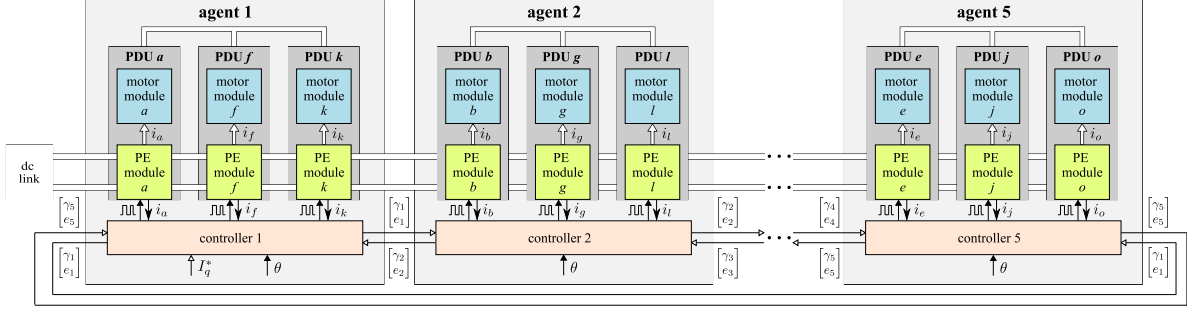


Fig. 1. MMD architecture with $n = 15$ PDUs grouped into $p = 5$ agents. Only agents 1 and 3 receive the total current set-point I_q^* (corresponding to a total torque demand) from the user.

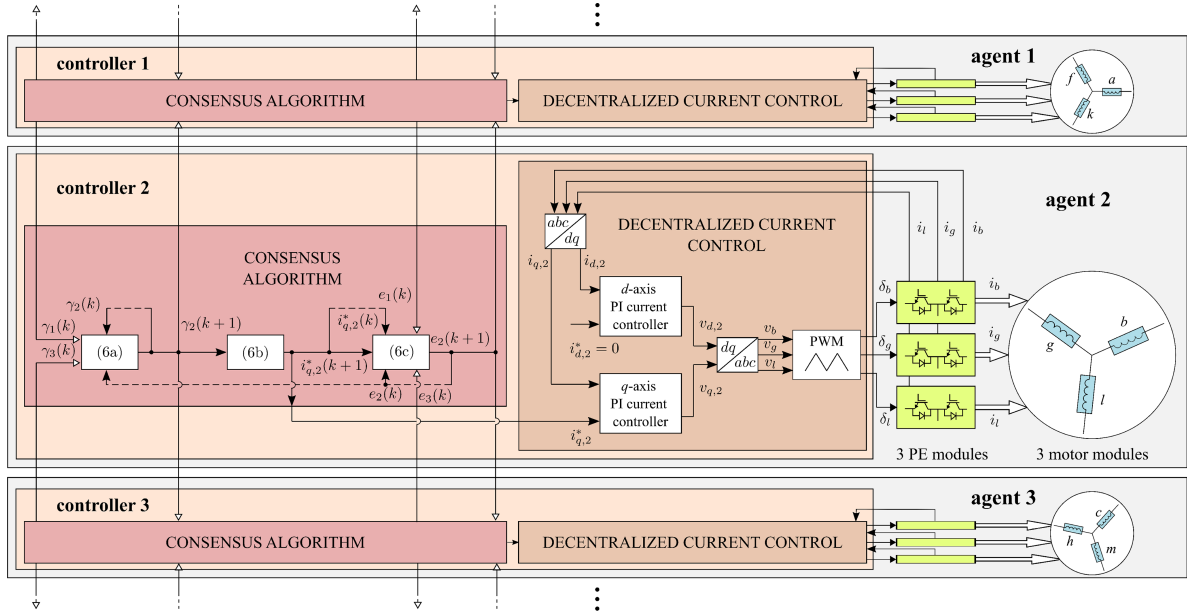


Fig. 2. Detail of agent 2 of Fig. 1. In general, each agent x communicates the parameters γ_x and e_x with its two closest neighbors (which are agent 1 and agent 3 for agent 2) to compute its agent-specific current set-point $i_{q,x}^*$ by means of the deterministic static average consensus algorithm of Section III. This set-point $i_{q,x}^*$ is then fed to the independent, three-phase field oriented current controller of that agent, which is elaborated upon in Section IV. This decentralized current controller requires only information of agent x itself, and determines the duty cycles for the three half-bridge inverter legs, which feed the three concentrated windings of this agent x . A three-phase current measurement provides feedback to the decentralized current controller.



Fig. 3. Communication network for (a) decentralized control with communication between closest neighbors only, and (b) centralized control with communication between the agents and the supervisory (SV) controller only. A * indicates the ability to communicate with the user.

field-oriented current controller of each agent regulates its stator currents to its set-points as explained in Section IV, without the need for additional communication with its neighbors.

III. DETERMINISTIC STATIC AVERAGE CONSENSUS ALGORITHM

The goal of the deterministic static average consensus algorithm is to let the agents decide among themselves which

agent has to deliver which part of a total torque demand. The agents can only use local computations and local interactions for this purpose, to comply with the flexibility and reliability of an MMD. A major advantage of the use of the consensus algorithm explained in this section, is the fact that the present condition of each agent can be taken into account during the distribution process.

For a surface-mounted PMSM under field-oriented control, the total torque demand T_{em}^* is directly proportional to a total current reference I_q^* : $T_{em}^* = 3N_p\Psi_d I_q^*/2$, with N_p the number of pole pairs and Ψ_d (Wb) the flux linkage caused by the permanent magnets. The distribution of a torque demand hence comes down to the distribution of a total current set-point I_q^* among the set-points $i_{q,x}^*$ of the individual agents, satisfying

$$\sum_{x=1}^p i_{q,x}^* = I_q^*. \quad (1)$$

Since no flux-weakening is considered in this article, $i_{d,x}^* = 0$ A for all agents x . For an interior PMSM, on the other hand, it is easier to divide the total torque demand T_{em}^* directly into multiple torque set-points $t_{em,x}^*$ for the individual agents. For every agent, the optimal current set-points $i_{q,x}^*$ and $i_{d,x}^*$ can then be computed using the maximum-torque per ampere profiles of the machine [33]. This allows to use a similar consensus algorithm as the one explained in this work (with I_q^* replaced by T_{em}^* , and $i_{q,x}^*$ replaced by $t_{em,x}^*$), but will not be elaborated specifically.

In order to obtain an allocation of the torque demand among the agents that takes into account the present state of each agent, a cost C_x is assigned to each agent. This cost C_x can be any quadratic function [29], [31], but in this article, it is chosen that

$$C_x = R_{s,x} i_{q,x}^{*2} \quad (2)$$

with $R_{s,x}$ (in Ω) the stator resistance of agent x . After all, a higher current set-point $i_{q,x}^*$ results in higher copper losses in agent x . This agent-specific cost function allows to distribute I_q^* in an unequal way over the p agents, depending on the differences in $R_{s,x}$, which reflect the present conditions of the agents. Differences in $R_{s,x}$ can arise, for instance, from temperature differences or production tolerances, or from a fault in a PDU of an agent. Other losses in the drive, such as for instance the switching losses, can be taken into account in C_x as well. However, this is not elaborated upon in this article.

The goal of the consensus algorithm is to reach agreement on the so-called incremental cost γ_x , defined as

$$\gamma_x = \frac{dC_x}{di_{q,x}^*} = 2R_{s,x} i_{q,x}^*. \quad (3)$$

This incremental cost indicates for each agent x individually how (dis)advantageous it would be to decrease or increase its current set-point $i_{q,x}^*$, based on the knowledge of the current state of that agent. A fair share of I_q^* among the agents is hence obtained when the incremental costs are equal for all the agents $x \in \{1, \dots, p\}$

$$\gamma_1 = \gamma_2 = \dots = \gamma_x = \dots = \gamma_p = \gamma^* = \frac{I_q^*}{r} \quad (4)$$

with $r = \sum_{x=1}^p 1/(2R_{s,x})$. An agent with a higher temperature (and thus a higher stator resistance) will receive a lower current set-point, which will reduce its copper losses and thus its temperature.

Finally, also the current constraints in the machine must be respected by the consensus algorithm

$$i_{q,\min,x} \leq i_{q,x}^* \leq i_{q,\max,x} \quad (5)$$

with $i_{q,\min,x}$ and $i_{q,\max,x}$, respectively, the minimum and maximum allowable current in the PDUs of agent x . These agent-specific current constraints can be used to represent the present condition of an agent as well. When an open-circuit fault is detected in a motor- or PE module of agent f (for instance with the technique explained in [34]), it suffices to set the current constraints $i_{q,\min,f}$ and $i_{q,\max,f}$ equal to zero to represent this faulty state.

A. Algorithm Description

To make sure that the total current reference I_q^* is delivered, i.e., to satisfy constraint (1), each agent x makes a local estimate e_x of the mismatch between the total required current I_q^* and the total delivered current $\sum i_{q,x}^*$. The consensus algorithm is hence based on the three parameters $i_{q,x}^*$, γ_x , and e_x .

All the agents are assumed to have exchanged their parameters γ_x and e_x with their two closest neighbors in one update period $T_c = 1/f_c$ of the consensus algorithm, as is illustrated in Fig. 1. At each discrete time instant k (with a period T_c between subsequent time instances), the agents simultaneously update their state variables $i_{q,x}^*$, γ_x , and e_x by means of the following local computations, based on the communicated information [31]:

$$\gamma_x(k+1) = \frac{[\gamma_{x-1}(k) + \gamma_x(k) + \gamma_{x+1}(k)]}{3} + \varepsilon e_x(k) \quad (6a)$$

$$i_{q,x}^*(k+1) = f_x(\gamma_x(k+1)) \quad (6b)$$

$$e_x(k+1) = \frac{[e_{x-1}(k) + e_x(k) + e_{x+1}(k)]}{3} - [i_{q,x}^*(k+1) - i_{q,x}^*(k)] \quad (6c)$$

with $x-1$, x and $x+1$ ($\in \{1, \dots, p\}$) three neighboring agents (e.g., agents 1, 2, and 3), and

$$f_x(\gamma_x) = \begin{cases} i_{q,\min,x}, & \text{if } \gamma_x < \gamma_{\min,x} \\ \frac{\gamma_x}{2R_{s,x}}, & \text{if } \gamma_{\min,x} \leq \gamma_x \leq \gamma_{\max,x} \\ i_{q,\max,x}, & \text{if } \gamma_x > \gamma_{\max,x} \end{cases} \quad (7)$$

with $\gamma_{\min,x} = 2R_{s,x} i_{q,\min,x}$ and $\gamma_{\max,x} = 2R_{s,x} i_{q,\max,x}$. The parameter ε is the so-called learning gain, which is a sufficiently small positive constant. The design of ε is the topic of Section III-B. The left part of Fig. 2 provides a schematic overview of algorithm (6).

The parameters γ_x , $i_{q,x}^*$, and e_x are initialized as follows:

$$\begin{cases} i_{q,x}^*(0) = 0 \\ \gamma_x(0) = 2R_{s,x} i_{q,x}^*(0) = 0 \\ e_x(0) = \begin{cases} \frac{I_q^*(0)}{|\mathcal{U}|} - i_{q,x}^*(0) = \frac{I_q^*(0)}{|\mathcal{U}|}, & \text{if } x \in \mathcal{U} \\ -i_{q,x}^*(0) = 0, & \text{otherwise.} \end{cases} \end{cases} \quad (8)$$

The subgroup $\mathcal{U}$ comprises the agents that can receive the total current set-point I_q^* directly from the user. $|\mathcal{U}|$ is the number of agents in this subgroup. For the case study used as an example throughout this article, only agents 1 and 3 can receive I_q^* from the user, i.e., $\mathcal{U} = \{1, 3\}$ and $|\mathcal{U}| = 2$.

When the total current reference I_q^* is changed at time instant k_0 , the estimated mismatches $e_x(k_0)$ of the agents $x \in \mathcal{U}$ are updated as follows:

$$e_x(k_0) = e_x(k_0) + [I_q^*(k_0) - I_q^*(k_0 - 1)] / |\mathcal{U}|. \quad (9)$$

The basic idea of algorithm (6) is that each agent locally estimates the mismatch between the demanded current I_q^* on the one hand, and the total delivered current $\sum i_{q,x}^*$ on the other hand, by means of the parameter e_x . The agents share this information with their two closest neighbors. Due to this communication [which is expressed mathematically in the term

$[e_{x-1} + e_x + e_{x+1}]/3$ of (6c)], the information about I_q^* —which is initially only known by the agents which are part of the subgroup $\mathcal{U}$ according to (9)—is spread out over all the agents. Assume for instance, without loss of generality, that I_q^* increases. Update (9) and the term $[e_{x-1} + e_x + e_{x+1}]/3$ in (6c) eventually cause all estimated mismatches to increase. The term εe_x in (6a) on his turn increases the incremental costs γ_x of all the agents, resulting in an increase in $i_{q,x}^*$ according to (6b). The term $[-i_{q,x}^*(k+1) - i_{q,x}^*(k)]$ in (6c) prevents that $i_{q,x}^*$ keeps increasing when $\sum i_{q,x}^*$ matches I_q^* again. The exchange of the incremental cost, which is expressed mathematically by means of the term $[\gamma_{x-1} + \gamma_x + \gamma_{x+1}]/3$ in (6a), results in consensus on the incremental cost, i.e., the satisfaction of constraint (4).

B. Learning Gain Design

The learning gain ε strongly influences the convergence speed of algorithm (6). From the proof of convergence of similar algorithms in [25], [29], and [31] can be concluded that the fastest convergence can be obtained with the learning gain ε that minimizes the in modulus second largest eigenvalue $|\lambda_2|$ of the system matrix $M \in \mathbb{R}^{2p \times 2p}$ of the consensus algorithm. This system matrix M can be obtained by rewriting algorithm (6) in the following form:

$$\begin{bmatrix} \gamma(k+1) \\ e(k+1) \end{bmatrix} = \underbrace{\begin{bmatrix} P & \varepsilon I \\ R(I-P) & P - \varepsilon R \end{bmatrix}}_M \begin{bmatrix} \gamma(k) \\ e(k) \end{bmatrix} \quad (10)$$

with γ and $e \in (\mathbb{R}^{p \times 1})$ the column vectors of γ_x and e_x , respectively, I the $p \times p$ identity matrix, and

$$R = \text{diag} \left[\frac{1}{2R_{s,1}}, \dots, \frac{1}{2R_{s,p}} \right].$$

The communication matrix $P \in \mathbb{R}^{p \times p}$ contains the elements p_{xy} that are equal to $1/3$ when agent x and y can communicate (i.e., when $x = y$, or when x and y are closest neighbors), and are zero otherwise. For the case study on the modular AFPMSM with $p = 5$ and communication between closest neighbors only [as shown in Fig. 3(a)], this leads to

$$P = \begin{bmatrix} 1/3 & 1/3 & 0 & 0 & 1/3 \\ 1/3 & 1/3 & 1/3 & 0 & 0 \\ 0 & 1/3 & 1/3 & 1/3 & 0 \\ 0 & 0 & 1/3 & 1/3 & 1/3 \\ 1/3 & 0 & 0 & 1/3 & 1/3 \end{bmatrix}.$$

It must be noted that the calculation of the optimal learning gain ε_{opt} requires global system information (including the resistances $R_{s,x}$ of all the agents, and the lay-out of the communication network). As a result, the learning gain ε needs to be determined offline, before the consensus algorithm starts running and only partial information is available to each agent. For the case study on the modular AFPMSM with stator winding resistances $R_{s,x} = 65 \text{ m}\Omega$ for all agents x and communication between closest neighbors only, minimization of the in modulus second largest eigenvalue $|\lambda_2|$ of M by means of a simple grid-search leads to the optimal learning gain $\varepsilon_{\text{opt}} = 0.0257$, as can be seen in Fig. 4. It must be noted, however, that the motor parameters

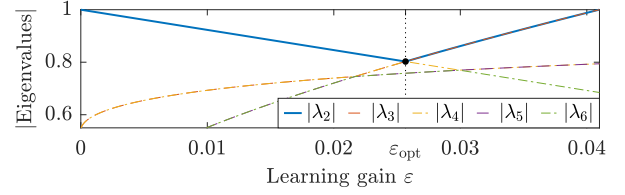


Fig. 4. Variation of the moduli of the eigenvalues $|\lambda_i| (i \in \{2, \dots, 6\})$ of system matrix M in function of the learning gain ε . The optimal learning gain ε_{opt} corresponds to the minimum of $|\lambda_2|$.

can vary during operation of the MMD. As the parameter ε_{opt} is determined offline, it will not take these changes into account. However, a fixed value of ε results in adequate performance for a range of stator winding resistances. For $\varepsilon = 0.0257$, for instance, a range of resistances $[0.7 \ 0.8 \ 0.9 \ 1 \ 1.1 \ 1.2 \ 1.3] \cdot 65 \text{ m}\Omega$ requires, respectively, $[23 \ 14 \ 12 \ 13.33 \ 15 \ 16.33 \ 17.67]$ iterations for convergence (averaged over a change ΔI_q^* of $+15 \text{ A}$, -25 A , and $+10 \text{ A}$). Convergence is reached when the following requirements are satisfied for all agents x :

$$\left| \frac{\sum_x i_{q,x}^* - I_q^*}{I_q^*} \right| < 0.05 \text{ and } \left| \frac{\gamma_x - \gamma^*}{\gamma^*} \right| < 0.05. \quad (11)$$

C. Case Study

The performance of consensus algorithm (6) will be simulated for three operation conditions.

- 1) Healthy operation conditions, under which the stator winding resistances $R_{s,x}$ and the current constraints $i_{q,\min,x} = 0 \text{ A}$ and $i_{q,\max,x} = 10 \text{ A}$ are the same for every agent x
- 2) Faulty operation conditions, under which an open-circuit fault is detected in a PE module of agent 1 (for instance by comparison of its measured phase current and its corresponding reference signal, as explained in [34]), resulting in $i_{q,\min,1} = i_{q,\max,1} = 0 \text{ A}$
- 3) Operation with different stator winding resistances $R_{s,x}$ for the different agents.

The first 30 iterations of Fig. 5 show the variation of the parameters $i_{q,x}^*$, e_x , and γ_x for agents 1, 2 and 4 in the case study for an increase in the total reference I_q^* from 0 to 20 A under healthy operation conditions and with identical stator winding resistances for each agent. The parameters of agents 3 and 5 vary very similarly and are therefore not shown. When I_q^* changes at iteration k , the parameters of agent 1 and 3 immediately adapt according to (9), because the change in I_q^* is directly communicated to these two agents. The communication between the agents themselves results in a reaction of the other agents at iteration $k+1$. The effect in agent 2 is stronger than the effect in agents 4 and 5, as agent 2 receives information from both agents 1 and 3. Eventually, the set-points $i_{q,x}^*$ of all the agents converge to $I_q^*/5 = 4 \text{ A}$, the estimated mismatches e_x converge to 0 A, and the incremental costs γ_x converge to $\gamma^* = I_q^*/(\sum_{x=1}^5 1/(2R_{s,x})) = 0.52 \text{ W/A}$. Convergence requirements (11) are fulfilled after 14 iterations.

At iteration 30 in Fig. 5, an open-circuit fault is detected in a PDU of agent 1, after which all the PE switches of agent 1

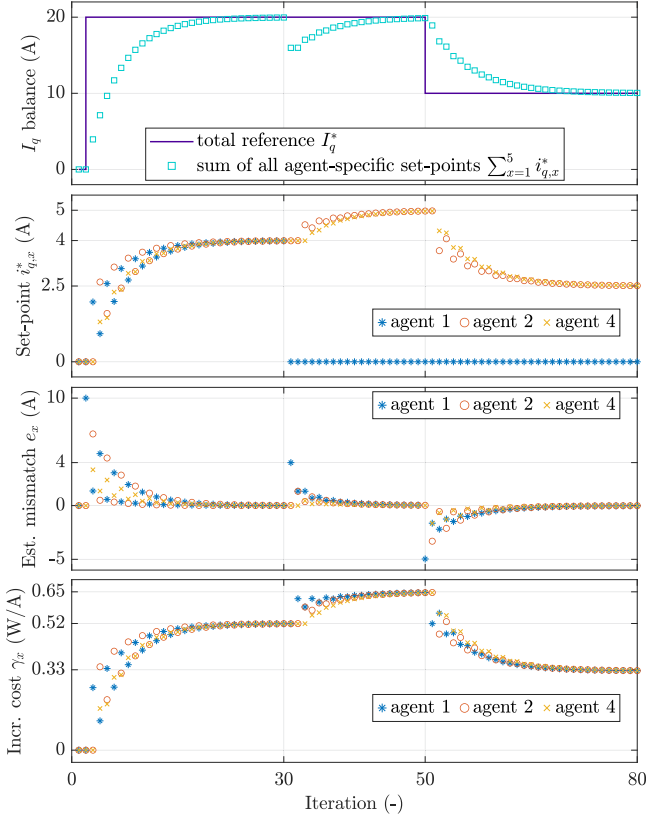


Fig. 5. Variables of the consensus algorithm of agent 1, 2, and 4 for a change in reference I_q^* at iteration 2 and 50, and an open-circuit fault in agent 1 at iteration 30 ($\varepsilon = \varepsilon_{\text{opt}}$).

are opened. The current constraints $i_{q,\min,1}$ and $i_{q,\max,1}$ are immediately set equal to zero, to take the faulty condition of agent 1 into account. It must be noted that there is no need for agent 1 to communicate its present condition to its neighbors, or to a master or supervisory controller. It suffices that agent 1 adapts its own parameters, after which the execution of algorithm (6) remains unchanged. The change in current constraints results in $i_{q,1}^*(31) = 0$ A and an increase in $e_1(31)$. This increase in $e_1(31)$ results in an increase in $\gamma_1(32)$. At iteration 32, the info about the increase in e_1 has reached agents 2 and 5 through communication with agent 1, causing a rise in e_2 and e_5 , which in turn results in an increase in γ_2 and γ_5 at iteration 33. The effect in agents 3 and 4 is only noticed at iteration 33, when they get info on the increase of e_2 and e_5 , respectively. The increased incremental costs allow an increase in the set-points $i_{q,x}^*$ of the healthy agents, until the total set-point I_q^* is reached again. After 10 iterations, the set-points of the healthy agents have converged to $I_q^*/4$, and all the agents agree on an incremental cost equal to $I_q^*/(\sum_{x=2}^5 1/(2R_{s,x}))$. Even under these faulty conditions, I_q^* can still be varied, as is shown at iteration 50.

The variation of the parameters $i_{q,x}^*$, e_x , and γ_x for the situation in which the agents have different stator winding resistances (i.e., $R_{s,1} = 58.5 \Omega$, $R_{s,2} = 71.5 \Omega$, $R_{s,3} = 97.5 \Omega$, $R_{s,4} = 45.5 \Omega$, and $R_{s,5} = 65.0 \Omega$) is depicted in Fig. 6. After 14 iterations, the agents agree on an incremental cost $\gamma^* = I_q^*/(\sum_{x=1}^5 1/(2R_{s,x})) = 0.51$ W/A, which results in different

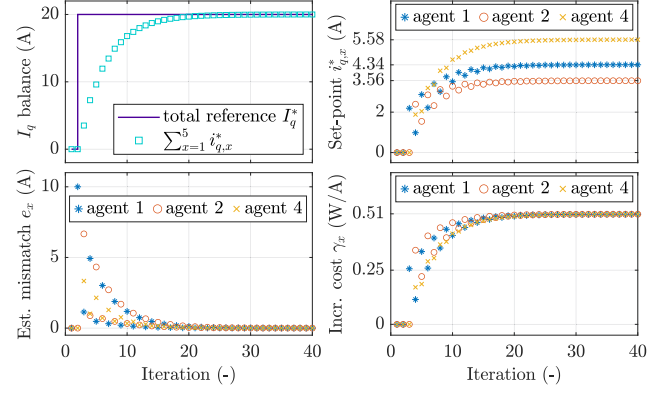


Fig. 6. Variables of the consensus algorithm of agent 1, 2, and 4 for a change in demand I_q^* at iteration 2 when the agents have different stator winding resistances ($\varepsilon = \varepsilon_{\text{opt}}$).

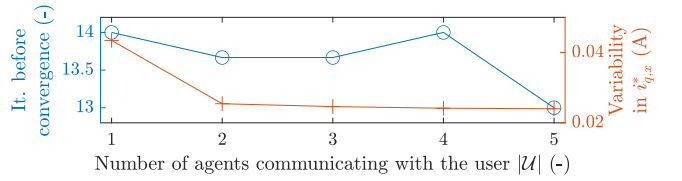


Fig. 7. Effect of the number of agents $|U|$ that can communicate directly with the user on the number of iterations (it.) that is required for convergence, and on the variability (var.) in $i_{q,x}^*$ ($\varepsilon = \varepsilon_{\text{opt}}$).

set-points $i_{q,x}^* = \gamma^*/(2R_{s,x})$ for the different agents. The torque is hence distributed unequally among the agents, but the total torque demand is still met, as $\sum_x i_{q,x}^*$ converges to I_q^* .

D. Effect of the Number of Agents Able to Communicate With the User

Throughout this article, it is assumed that only agents 1 and 3 can communicate with the user, and hence immediately receive the set-point I_q^* . The number of agents $|U|$ that can communicate with the user is a tradeoff between redundancy on the one hand, and additional communication links and communication load on the other hand. Fig. 7 shows the effect of $|U|$ on the performance of the consensus algorithm for the case study with five identical agents, i.e., the effect of $|U|$ on the number of iterations that is required for convergence, and on the variability (and hence the ripple) in $i_{q,x}^*$. Convergence is reached when requirements (11) are satisfied for all agents. The variability in $i_{q,x}^*$ is defined as

$$\frac{\sum_k |i_{q,x}^*(k) - i_{q,x}^*(k-1)|}{\text{\#steps}}$$

with #steps the number of changes observed in $i_{q,x}^*$. Both quantities are averaged over a change ΔI_q^* of +15 A, -25 A, and +10 A, and over all the possibilities for a specific $|U|$. $|U|$ does not have a large effect on the convergence speed. According to (9), a smaller number $|U|$ results in a larger change in e_y for the agents $y \in U$, which in turn results in more abrupt changes in the parameters e_z , γ_z , and $i_{q,z}^*$ of the other agents z after the communication of e_y . A larger number $|U|$, on the other hand, results in a smaller change in e_y per agent in U , but affects more agents at once. This phenomenon also explains why the variability in $i_{q,x}^*$ is higher for smaller $|U|$.

E. Comparison With a Simple Master–Slave and a Supervisory Control Approach

In a simple master–slave control approach, one of the agents of the MMD is designated the role of master controller that continuously keeps track of the number of agents p in the system. This master controller divides the reference I_q^* into p equal set-points $i_q^* = I_q^*/p$, after which this set-point is communicated from neighbor to neighbor to all the agents by means of the same communication network as the one used for the consensus algorithm [see Fig. 3(a)]. In this way, the total current set-point I_q^* is divided among the agents in $\lfloor p/2 \rfloor$ iterations.

Another option is to use one central, supervisory controller to distribute I_q^* among the agents, which can communicate with all the agents in the system to gather global system information [see Fig. 3(b)]. This supervisory controller can compute the optimal incremental cost $\gamma^* = I_q^*/(\sum_{x=1}^p 1/(2R_{s,x}))$ instantaneously, together with the set-points $i_{q,x}^* = \gamma^*/(2R_{s,x})$ for each agent x [29].

Both the simple master–slave control approach and the supervisory controller are able to split I_q^* in the set-points $i_{q,x}^*$ in a faster and a computationally more efficient way than the consensus algorithm of Section III-A. However, the simple master–slave control approach is not able to take into account the present conditions of the agents (such as differences in $R_{s,x}$ due to for instance temperature differences or faults), nor to distribute I_q^* in an unequal way over the p agents, while Fig. 6 confirms that the consensus algorithm is able to deal with this situation. The supervisory controller, on the other hand, is in complete disagreement with the modular approach in an MMD.

IV. DECENTRALIZED CURRENT CONTROL

A. Algorithm Description

To complete the multiagent current control strategy, the consensus algorithm explained in Section III is combined with a decentralized control strategy, in which each agent gets an independent m -phase PI controller. For the case study used throughout this research, this means that each of the five agents is equipped with an independent, standard 3-phase PI controller to reach the individual set-points $i_{q,x}^*$ and $i_{d,x}^* = 0$ A generated by consensus algorithm (6), as is illustrated in Fig. 2. The PI parameters are tuned according to the state-of-the-art procedure presented in [11], which takes into account the internal magnetic coupling between the n motor modules, represented by the mutual inductances tabulated in Table I.

B. Case Study

By carrying out a multivariable frequency analysis according to [11], the PI parameters are chosen in such a way that a set-point amplitude change $\Delta i_{q,x}^*$ in agent x does not deviate the currents $i_{q,y}$ and $i_{d,y}$ in the other agents y with more than 13% of the magnitude of $\Delta i_{q,x}^*$. The PI parameters of each agent are set equal to $K_p = 2$ and $K_i = 200$, resulting in a natural frequency $\omega_n = 803.28$ rad/s and a damping factor $\zeta = 4.15$. The simulation results under a mechanical speed N of 400 r/min, with a dc-bus voltage V_{dc} of 48 V, a sample frequency f_s of 10 kHz, a consensus update frequency f_c of 2 kHz, and the

TABLE I
MACHINE PARAMETERS

Parameter	Symbol	Value
Rated torque (N.m)	T_n	15
Rated speed (r/min)	N_n	2500
Number of pole pairs	N_p	8
Stator winding resistance (mΩ)	$R_{s,x} (\forall x \in \{1, \dots, 5\})$	65
Self- and mutual inductances (μH) ($s \in \{1, 2, \dots, 15\}$)	$L_{s,s}$	316
	$L_{s,s-1} = L_{s,s+1}$	111
	$L_{s,s-2} = L_{s,s+2}$	15.5
	$L_{s,s-3} = L_{s,s+3}$	9.52
	$L_{s,s-4} = L_{s,s+4}$	8.27
	$L_{s,s-5} = L_{s,s+5}$	6.05
	$L_{s,s-6} = L_{s,s+6}$	6.07
	$L_{s,s-7} = L_{s,s+7}$	5.31
Flux linkage caused by magnets (Wb)	Ψ_d	0.02

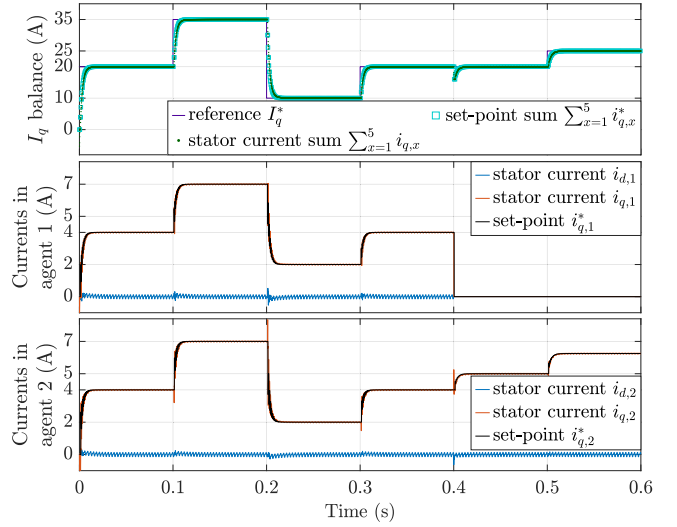


Fig. 8. Simulation results for $\varepsilon = \varepsilon_{\text{opt}}$, $N = 400$ r/min, $V_{dc} = 48$ V, $f_s = 10$ kHz, $f_c = 2$ kHz. The agents are identical. An open-circuit fault occurs in agent 1 at $t = 0.4$ s. The currents in agents 3, 4, and 5 are very similar to the ones in agent 2, and are therefore not shown explicitly. The combination of the consensus algorithm and the decentralized PI current controllers ensures that the total current reference is met, i.e., $I_q^* = \sum_{x=1}^5 i_q^* = \sum_{x=1}^5 i_{q,x}$.

optimal learning gain $\varepsilon_{\text{opt}} = 0.0257$ are depicted in Fig. 8 for the situation in which all five agents are identical. One iteration of consensus algorithm (6), including all the communications between the agents, is executed in one consensus update period $T_c = 1/f_c$. One iteration of the PI controller, including the measurement of the phase currents through the motor modules and the generation of the pulses for the PE modules, is executed in one sample period $T_s = 1/f_s$. The simulation results show that the consensus algorithm indeed results in an adequate distribution of the total current reference I_q^* (i.e., $\sum_x i_{q,x}^* = I_q^*$), even after an open-circuit failure in agent 1 after 0.4 s. The decentralized PI controllers of the different agents are able to follow their set-points $i_{q,x}^*$ well, causing the total current set-point—and hence the total torque demand—to be reached (i.e., $\sum_x i_{q,x} = I_q^*$).

V. EXPERIMENTAL VALIDATION

The simulation results of Section IV are validated on the 4 kW SAT axial-flux PM synchronous machine test setup of Fig. 9, with the specifications listed in Table I. The AFPMSM

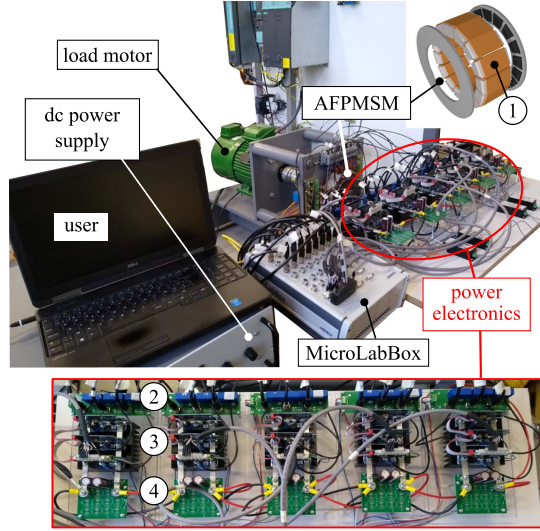


Fig. 9. Test setup consisting of a load motor and a modular AFPMSM with 15 motor modules (1), which are grouped into five agents. The user can communicate the set-point I_q^* to the dSPACE MicroLabBox, in which the five controllers of the agents are implemented in a completely separate way. The phase currents are measured with five sets of three current sensors (2). Each three-phase agent comprises a dedicated three-phase power module (3) with its own dc-link connection (4). All power modules are connected in parallel to a dc power supply.

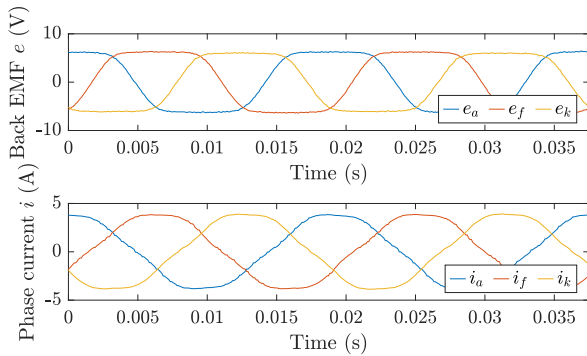


Fig. 10. Measured three-phase back EMF and current waveforms for agent 1 of the modular AFPMSM ($N = 400$ r/min, $V_{dc} = 48$ V, $f_s = 10$ kHz).

is working in motor mode, while its mechanical speed of 400 r/min is maintained by means of an induction motor drive. A Lorenz DR-2112 torque transducer with a measurement range of ± 50 N · m and an accuracy of 0.1% is mounted on the shaft between this AFPMSM and this induction motor. The AFPMSM consists of 15 identical motor modules, of which all 30 power terminals are accessible. The motor modules are grouped into five sets of three modules connected in one neutral point, and fed by a modular low voltage scalable power platform from Infineon. As can be seen in Fig. 9, this power platform comprises five sets of three PE modules. An agent hence includes three motor modules connected to three PE modules, in combination with a set of three LEM LA 25-P current transducers. The five agents are connected in parallel to a dc power supply with $V_{dc} = 48$ V. The experimentally obtained three-phase back EMF and current waveforms for agent 1 are shown in Fig. 10. The back

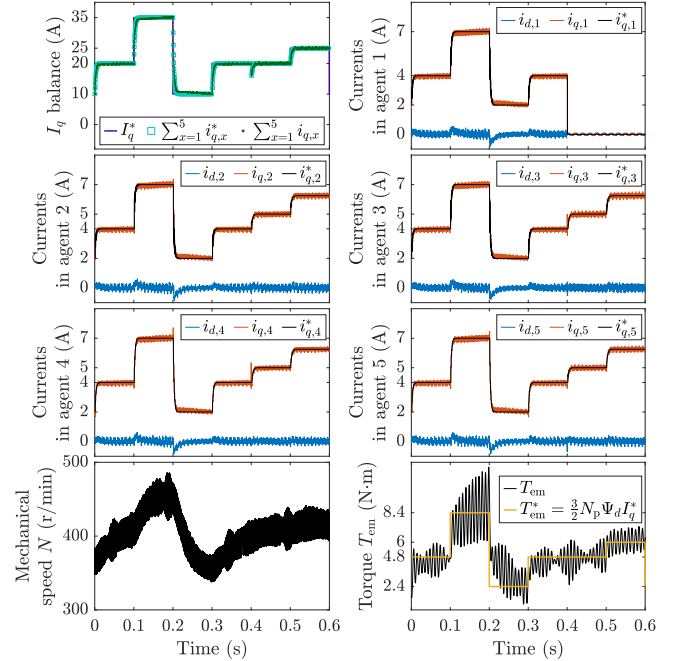


Fig. 11. Experimental results for $\varepsilon = \varepsilon_{opt}$, $V_{dc} = 48$ V, $f_s = 10$ kHz, $f_c = 2$ kHz. The agents are identical. An open-circuit fault occurs in agent 1 at $t = 0.4$ s.

EMF waveforms are obtained by means of three Cleverscope CS1030 differential probes, while the phase current waveforms are obtained by means of the LEM LA 25-P current transducers.

The consensus algorithm of Section III and decentralized PI control of Section IV are implemented on the FPGA of a single dSPACE MicroLabBox, in order to be able to freely vary the communication frequency and hence the consensus update frequency f_c , without the limitations of a specific communication protocol. The effect of f_c will be discussed in Section VI. Although all controllers are implemented on the same dSPACE MicroLabBox, they are implemented in a completely separate way, allowing only the transfer of γ_x and e_x between the controllers of neighboring agents at a variable communication frequency, as is illustrated in Fig. 1. The stator current measurements $i_{q,x}$ and $i_{d,x}$ are obtained from the fifteen LEM LA 25-P current transducers providing feedback to the PI controllers at a sampling rate f_s of 10 kHz. The measurement results are drawn and recorded from the MicroLabBox by means of ControlDesk software at the same sampling rate of 10 kHz.

The obtained experimental results shown in Fig. 11 for five identical agents are in very good agreement with the simulation results of Fig. 8. However, the steps in the torque demand T_{em}^* (and hence in the current reference I_q^*) cause speed fluctuations in the experimental set-up, which were not taken into account in the simulations. But even with these speed fluctuations, the suggested decentralized multiagent current control approach (which is the combination of the consensus algorithm of Section III and the decentralized PI control of Section IV) indeed makes sure that the total torque demand is met (i.e., $\sum_x i_{q,x} = I_q^*$), even under an agent malfunction. Fig. 12 shows the variation of the current and torque ripple under variable speed. It can be noticed

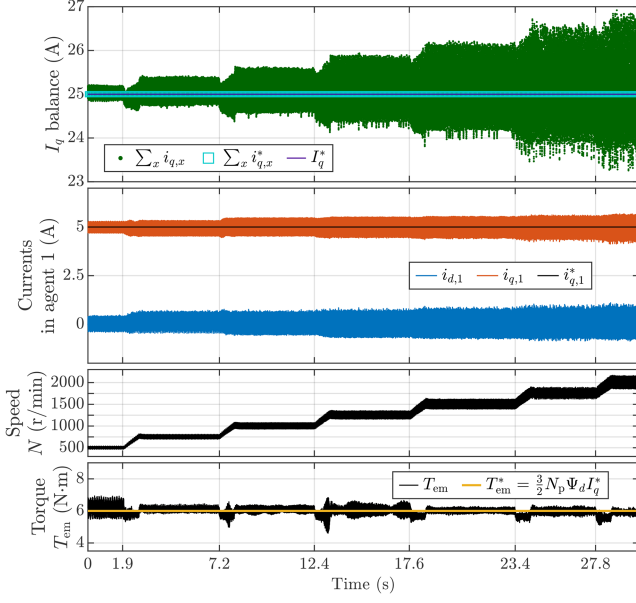


Fig. 12. Experimentally obtained current- and torque waveforms under variable speed ($\varepsilon = \varepsilon_{\text{opt}}$, $V_{\text{dc}} = 100$ V, $f_s = 10$ kHz). The agents are identical.

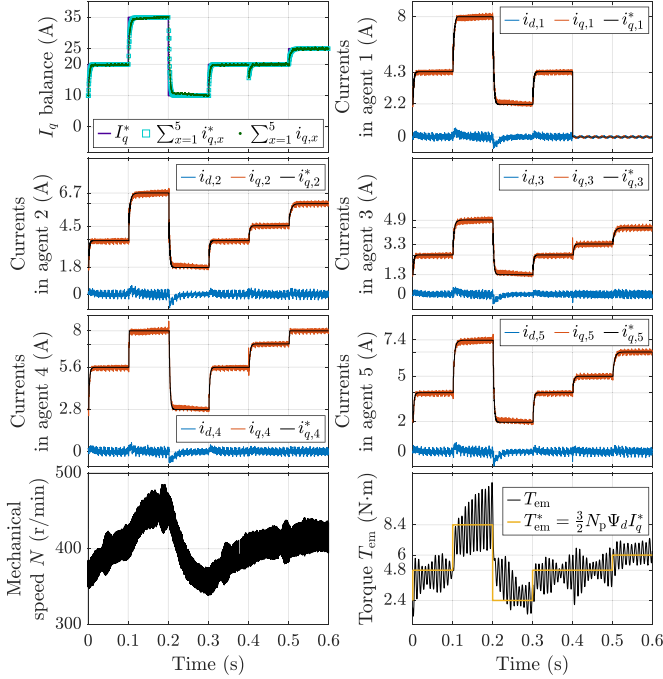


Fig. 13. Experimental results for nonidentical agents ($\varepsilon = \varepsilon_{\text{opt}}$, $V_{\text{dc}} = 48$ V, $f_s = 10$ kHz, $f_c = 2$ kHz). The agents use different stator winding resistances in their cost function (2). An open-circuit fault occurs in agent 1 at $t = 0.4$ s.

that the current ripple increases with increasing speed. However, even at a mechanical speed of 2000 r/min, stator current component $i_{q,1}$ of agent 1 is retained between 4.1 and 5.7 A for a set-point $i_{q,1}^* = 5$ A, while the stator current component $i_{d,1}$ is retained between -0.9 A and 1.1 A for a set-point $i_{d,1}^* = 0$ A.

When the agents use different values for their stator winding resistances $R_{s,x}$ (i.e., $R_{s,1} = 58.5 \Omega$, $R_{s,2} = 71.5 \Omega$, $R_{s,3} =$

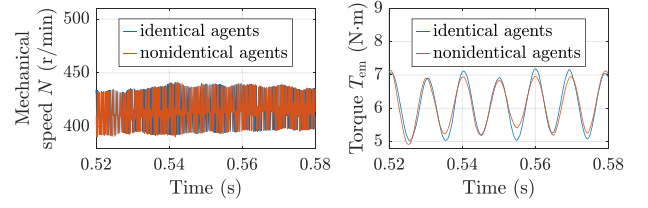


Fig. 14. Detail of the speed and torque waveforms shown in Figs. 11 and 13 (identical agents) and (nonidentical agents).

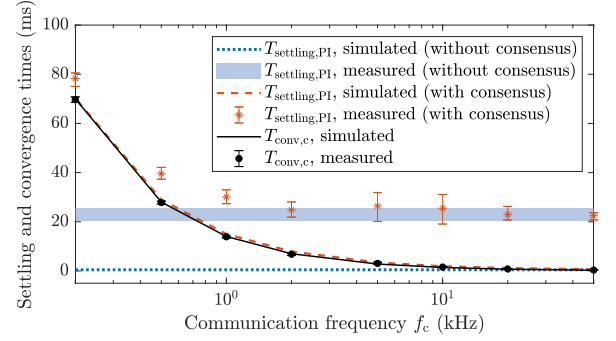


Fig. 15. Simulated and measured PI settling time $T_{\text{settling,PI}}$ and consensus convergence time $T_{\text{conv,c}}$ in function of consensus update frequency f_c ($\varepsilon = \varepsilon_{\text{opt}}$, $N = 400$ r/min, $V_{\text{dc}} = 48$ V, $f_s = 10$ kHz). The measured results are averaged over at least 10 experiments, and the error bars show the minimum, mean, and maximum of the obtained results.

97.5Ω , $R_{s,4} = 45.5 \Omega$, and $R_{s,5} = 65.0 \Omega$) in their cost function (2), the total torque demand—and hence the total current reference I_q^* —is distributed unevenly among the agents of the MMD, as is shown in Fig. 13. The decentralized PI current controllers of Section IV are still able to follow these agent-specific current set-points, even under an agent malfunction. The measured torque ripple and speed fluctuations under this unequal torque distribution are similar to the ones under the equal torque distribution, as can be seen in Fig. 14, which shows these fluctuations over three electrical periods. The torque oscillates at twice the stator electrical frequency, due to the interaction of the third harmonic component in the back EMF and the fundamental current. The feature that the suggested decentralized multiagent control approach can deal with differences between the agents without the need for a supervisory controller, is one of its major advantages.

VI. INFLUENCE OF CONSENSUS UPDATE FREQUENCY ON CONVERGENCE AND SETTLING TIME

It is important to notice that the update frequency f_c of the consensus algorithm of Section III does not need to be related with the sample frequency f_s of the decentralized PI current controllers of Section IV: f_c can be higher or lower than f_s . Fig. 15 shows the simulated and experimentally obtained PI settling time $T_{\text{settling,PI}}$ (ms) and consensus convergence time $T_{\text{conv,c}}$ (ms), averaged over a change ΔI_q^* in total set-point of +15, -25, and +10 for different choices of f_c for five identical agents. These two parameters are defined as the time it takes to fulfill $|\sum_x i_{q,x} - I_q^*| < 0.05$ for $T_{\text{settling,PI}}$; and requirements (11)

for $T_{\text{conv},c}$. On average, (11) is fulfilled after 13.33 iterations, and hence $T_{\text{conv},c} = 13.33/f_c$. The simulated and measured PI settling times for the situation without consensus algorithm, i.e., where each agent x immediately receives $i_{q,x}^* = I_q^*/p$ as set-point from the supervisory controller discussed in Section III-E, are added as reference values in Fig. 15.

In simulation, both $T_{\text{settling,PI}}$ and $T_{\text{conv},c}$ vary inversely proportional to the consensus update frequency f_c . $T_{\text{settling,PI}}$ is always slightly larger than $T_{\text{conv},c}$. In the measurements, however, the PI settling times are consistently higher than in simulation. This is due to the speed disturbances that occur when I_q^* (and hence the torque) is changed, which are not taken into account in the simulations.

The rate $|d(\Delta I_q^*)/dt|$ at which the total set-point I_q^* can vary, is hence inversely proportional to f_c . As a result, the requirements of the application can impose a lower bound on f_c . The type of communication protocol, on the other hand, will impose an upper bound on f_c . If, for instance, a multimaster CAN communication network is used—which has the advantage that all the nodes in the network are truly peers—the maximum bit rate is limited to 1 Mb/s [35]. If all incremental costs γ_x and estimated mismatches e_x are represented by means of 16 b, $5 \cdot 2 \cdot 16 = 160$ b must be transmitted over the network for one iteration of the consensus algorithm (6). As a result, f_c will be limited to 6.25 kHz. For this particular case study, in which I_q^* changes every 100 ms, this is not a severe limitation: the simulation and experimental result of Figs. 8 and 11 show that a consensus update frequency f_c of 2 kHz is already sufficient to achieve adequate results. In this case, the consensus algorithm has already converged in 7 ms. Furthermore, the experimental results in Fig. 15 show that, in the measurements, the PI settling time is almost not affected anymore by the consensus update frequency f_c when $f_c > 2$ kHz: for $f_c > 2$ kHz, it is the dynamic performance of the complete drive that limits the allowable $|d(\Delta I_q^*)/dt|$, and not the consensus algorithm.

VII. CONCLUSION

In this article, a multiagent control strategy applicable to MMDs was introduced. The MMD was subdivided into different agents, each equipped with a dedicated controller. A deterministic static average consensus algorithm was implemented in each agent, to let the agents decide among themselves, which agent had to deliver which part of a total torque demand, taking into account the present condition of each agent, but without the need for a central controller. The autonomous PI current controllers of the agents then regulated the phase currents toward the agent-specific current set-points obtained in this way. Both simulations and experiments proved that the total torque demand was met, even under an open-circuit fault in one of the agents and under differences between the agents. The influence of the update frequency of the consensus algorithm on the consensus convergence time and the PI settling time was studied as well, providing useful insights in the allowable rate at which the total torque demand could change and the feasible communication protocols.

REFERENCES

- [1] A. Shea and T. M. Jahns, "Hardware integration for an integrated modular motor drive including distributed control," in *Proc. IEEE Energy Convers. Congr. Expo.*, 2014, pp. 4881–4887.
- [2] M. Uğur and O. Keysan, "Design of a GaN based integrated modular motor drive," in *Proc. XIII Int. Conf. Elect. Mach.*, 2018, pp. 1471–1477.
- [3] J. Wang, Y. Li, and Y. Han, "Integrated modular motor drive design with GaN power FETs," *IEEE Trans. Ind. Appl.*, vol. 51, no. 4, pp. 3198–3207, Jul./Aug. 2015.
- [4] R. Abebe *et al.*, "Integrated motor drives: State of the art and future trends," *IET Electric Power Appl.*, vol. 10, no. 8, pp. 757–771, 2016.
- [5] L. Verkroost, J. Van Damme, D. V. Bozalakov, F. De Belie, P. Sergeant, and H. Vansompel, "Simultaneous DC-link and stator current ripple reduction with interleaved carriers in multiphase controlled integrated modular motor drives," *IEEE Trans. Ind. Electron.*, vol. 68, no. 7, pp. 5616–5625, Jul. 2021.
- [6] M. D. Hennen, M. Niessen, C. Heyers, H. J. Brauer, and R. W. De Doncker, "Development and control of an integrated and distributed inverter for a fault tolerant five-phase switched reluctance traction drive," *IEEE Trans. Power Electron.*, vol. 27, no. 2, pp. 547–554, Feb. 2012.
- [7] A. H. Mohamed, H. Vansompel, and P. Sergeant, "An integrated modular motor drive with shared cooling for axial flux motor drives," *IEEE Trans. Ind. Electron.*, vol. 68, no. 11, pp. 10467–10476, Nov. 2021.
- [8] I. Zoric, M. Jones, and E. Levi, "Arbitrary power sharing among three-phase winding sets of multiphase machines," *IEEE Trans. Ind. Electron.*, vol. 65, no. 2, pp. 1128–1139, Feb. 2018.
- [9] G. Sala, M. Mengoni, G. Rizzoli, L. Zarri, and A. Tani, "Decoupled d - q axes current-sharing control of multi-three-phase induction machines," *IEEE Trans. Ind. Electron.*, vol. 67, no. 9, pp. 7124–7134, Sep. 2020.
- [10] G. Sala, G. Valente, M. Di Nardo, M. Degano, P. Zanchetta, and C. Gerada, "Power-sharing control in bearingless multi-sector and multi-three-phase permanent magnet machines," *IEEE Trans. Ind. Electron.*, vol. 68, no. 10, pp. 9070–9080, Oct. 2021.
- [11] E. Prieto-Araujo, A. Junyent-Ferré, D. Lavèrnia-Ferrer, and O. Gomis-Bellmunt, "Decentralized control of a nine-phase permanent magnet generator for offshore wind turbines," *IEEE Trans. Energy Convers.*, vol. 30, no. 3, pp. 1103–1112, Sep. 2015.
- [12] S. Rubino, O. Dordevic, R. Bojoi, and E. Levi, "Modular vector control of multi-three-phase permanent magnet synchronous motors," *IEEE Trans. Ind. Electron.*, vol. 68, no. 10, pp. 9136–9147, Oct. 2021.
- [13] W. Cao, B. C. Mecrow, G. J. Atkinson, J. W. Bennett, and D. J. Atkinson, "Overview of electric motor technologies used for more electric aircraft (MEA)," *IEEE Trans. Ind. Electron.*, vol. 59, no. 9, pp. 3523–3531, Sep. 2012.
- [14] M. J. Duran, I. González-Prieto, A. González-Prieto, and F. Barrero, "Multiphase energy conversion systems connected to microgrids with unequal power-sharing capability," *IEEE Trans. Energy Convers.*, vol. 32, no. 4, pp. 1386–1395, Dec. 2017.
- [15] A. Galassini, A. Costabeber, M. Degano, C. Gerada, A. Tassarolo, and R. Menis, "Enhanced power sharing transient with droop controllers for multithree-phase synchronous electrical machines," *IEEE Trans. Ind. Electron.*, vol. 66, no. 7, pp. 5600–5610, Jul. 2019.
- [16] J. W. Bennett, G. J. Atkinson, B. C. Mecrow, and D. J. Atkinson, "Fault-tolerant design considerations and control strategies for aerospace drives," *IEEE Trans. Ind. Electron.*, vol. 59, no. 5, pp. 2049–2058, May 2012.
- [17] Y. Han, K. Zhang, H. Li, E. A. A. Coelho, and J. M. Guerrero, "MAS-based distributed coordinated control and optimization in microgrid and microgrid clusters: A comprehensive overview," *IEEE Trans. Power Electron.*, vol. 33, no. 8, pp. 6488–6508, Aug. 2018.
- [18] X. Chen *et al.*, "Consensus-based distributed control for photovoltaic-battery units in a DC microgrid," *IEEE Trans. Ind. Electron.*, vol. 66, no. 10, pp. 7778–7787, Oct. 2019.
- [19] X. Chen *et al.*, "Distributed cooperative control of multiple hybrid energy storage systems in a DC microgrid using consensus protocol," *IEEE Trans. Ind. Electron.*, vol. 67, no. 3, pp. 1968–1979, Mar. 2020.
- [20] M. H. Ullah, B. Babaiahgari, A. Aleyat, and J.-D. Park, "A computationally efficient consensus-based multiagent distributed EMS for DC microgrids," *IEEE Trans. Ind. Electron.*, vol. 68, no. 6, pp. 5425–5435, Jun. 2021.
- [21] Z. Ding, "Consensus disturbance rejection with disturbance observers," *IEEE Trans. Ind. Electron.*, vol. 62, no. 9, pp. 5829–5837, Sep. 2015.
- [22] D. Meng and Y. Jia, "Robust consensus algorithms for multiscale coordination control of multivehicle systems with disturbances," *IEEE Trans. Ind. Electron.*, vol. 63, no. 2, pp. 1107–1119, Feb. 2016.

- [23] Y. Wang, Y. Song, and F. L. Lewis, "Robust adaptive fault-tolerant control of multiagent systems with uncertain nonidentical dynamics and undetectable actuation failures," *IEEE Trans. Ind. Electron.*, vol. 62, no. 6, pp. 3978–3988, Jun. 2015.
- [24] J. Qin, Q. Ma, Y. Shi, and L. Wang, "Recent advances in consensus of multi-agent systems: A brief survey," *IEEE Trans. Ind. Electron.*, vol. 64, no. 6, pp. 4972–4983, Jun. 2017.
- [25] K. Cai and H. Ishii, "Average consensus on general strongly connected digraphs," *Automatica*, vol. 48, no. 11, pp. 2750–2761, 2012.
- [26] S. S. Kia, B. Van Scoy, J. Cortes, R. A. Freeman, K. M. Lynch, and S. Martinez, "Tutorial on dynamic average consensus: The problem, its applications, and the algorithms," *IEEE Control Syst. Mag.*, vol. 39, no. 3, pp. 40–72, Jun. 2019.
- [27] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proc. IEEE*, vol. 95, no. 1, pp. 215–233, Jan. 2007.
- [28] F. Fagnani and P. Frasca, *Introduction to Averaging Dynamics Over Networks*, 1st ed. Berlin, Germany: Springer, 2018.
- [29] S. Yang, S. Tan, and J.-X. Xu, "Consensus based approach for economic dispatch problem in a smart grid," *IEEE Trans. Power Syst.*, vol. 28, no. 4, pp. 4416–4426, Nov. 2013.
- [30] T. Yang, D. Wu, Y. Sun, and J. Lian, "Minimum-time consensus-based approach for power system applications," *IEEE Trans. Ind. Electron.*, vol. 63, no. 2, pp. 1318–1328, Feb. 2016.
- [31] R. Wang, Q. Li, B. Zhang, and L. Wang, "Distributed consensus based algorithm for economic dispatch in a microgrid," *IEEE Trans. Smart Grid*, vol. 10, no. 4, pp. 3630–3640, Jul. 2019.
- [32] Y. Xu, S. Li, and J. Zou, "Integral sliding mode control based deadbeat predictive current control for PMSM drives with disturbance rejection," *IEEE Trans. Power Electron.*, vol. 37, no. 3, pp. 2845–2856, Mar. 2022.
- [33] T. M. Jahns, G. B. Kliman, and T. W. Neumann, "Interior permanent-magnet synchronous motors for adjustable-speed drives," *IEEE Trans. Ind. Appl.*, vol. IA/22, no. 4, pp. 738–747, Jul. 1986.
- [34] J. O. Estima and A. J. Marques Cardoso, "A new algorithm for real-time multiple open-circuit fault diagnosis in voltage-fed PWM motor drives by the reference current errors," *IEEE Trans. Ind. Electron.*, vol. 60, no. 8, pp. 3496–3505, Aug. 2013.
- [35] F. Savi, G. Buticchi, C. Gerada, P. Wheeler, and D. Barater, "Information technologies for distributed machine drives: An overview," in *Proc. IEEE Conf. Int. Electric Mach. Drives*, 2019, pp. 1805–1809.



Lynn Verkroost was born in Belgium, in 1994. She received the M.Sc. degree in electromechanical engineering in 2017 from Ghent University, Ghent, Belgium, where she is currently working toward the Ph.D. degree in electromechanical engineering.

Since 2017, she has been with the Electrical Energy Laboratory, Department of Electromechanical, Systems and Metal Engineering, Ghent University. She is an Affiliate Member of Flanders Make, the strategic research center

for the manufacturing industry in Flanders, Belgium. Her current research interests include digital, fault tolerant, and distributed control of converter-fed electrical machines.

Ms. Verkroost was the recipient of a Ph.D. fellowship from the Research Foundation-Flanders (FWO) in 2019.



Frederik De Belie was born in Belgium, in 1979. He received the M.Sc. and Ph.D. degrees in electromechanical engineering from Ghent University, Ghent, Belgium, in 2002 and 2010, respectively.

He is currently an Associate Professor with the Department of Electromechanical, Systems and Metal Engineering, Ghent University. He is also a Member of the Flanders Make Research Cluster on Control and Decisions. His research interests include identification, monitoring, dynamical modeling, and smart control theory applied to power converters and electrical drives.

Prof. De Belie was the recipient of the Iwan Åkerman Award for his Ph.D. work and patent on the seamless integration of sensorless drive in 2010, awarded on the initiative of Atlas Copco Airpower and Research Foundation-Flanders.



Peter Sergeant (Senior Member, IEEE) was born in Belgium, in 1978. He received the M.Sc. and Ph.D. degrees in electromechanical engineering from Ghent University, Ghent, Belgium, in 2001 and 2006, respectively.

In 2006, he became a Postdoctoral Researcher with Ghent University (Postdoctoral Fellow of the Research Foundation-Flanders). Since 2012, he has been an Associate Professor with Ghent University. He is also Core Lab Manager in the cluster Motion Products of

Flanders Make. His current research domain is electrical machines and drives for industrial and for sustainable energy applications. The focus is on accurate computation of losses in machines and drives, improving energy efficiency and increasing power density.



Hendrik Vansompel was born in Belgium, in 1986. He received the M.Sc. and Ph.D. degrees in electromechanical engineering from Ghent University, Ghent, Belgium, in 2009 and 2013, respectively.

He is currently a Postdoctoral Research Assistant and an Associate Professor with the Department of Electromechanical, Systems and Metal Engineering, Ghent University. He is an Affiliate Member of Flanders Make, the strategic research center for the manufacturing industry

in Flanders, Belgium. His research interests include electric machines and power electronics.