Adding Inverter Fault Detection to Model-Based Predictive Control for Flying-Capacitor Inverters

Joachim Druant, *Member, IEEE*, Thomas Vyncke, *Member, IEEE*, Frederik De Belie, *Member, IEEE*, Peter Sergeant *Member, IEEE*, and Jan Melkebeek *Member, IEEE*

***Abstract*—As inverters are often used in critical applications, reliability is an important issue. Especially the power electronic switches and gate drivers, the most essential components of the inverter, are vulnerable parts in real live operation. Therefore this paper focuses on open switch fault detection for multilevel inverters. When a single-switch open circuit fault occurs in one of the power electronic switches, the algorithm can detect the fault and the switch that is causing it. The detection is worked out for both a linear resistive inductive load and an induction motor. The proposed algorithm is an extension of an already available finite-set model based predictive control algorithm. Therefore no extra hardware or measurements are required. The paper also discusses a suggested method for reconfiguration after fault detection. Computer simulation and experimental verifications validate the proposed methods.**

***Index Terms*—Fault detection, MBPC, single-switch fault, FPGA implementation, induction machine**

I. Introduction

In the past few years there has been a tremendous interest in the model based predictive control of power electronics, as is clear from the large number of articles [1], [2], special sections in journals, books [3] and dedicated conference sessions [4].

Although many different approaches have been described, there seems to be some convergence on the use of finite-set model based predictive control (FS-MBPC) [5], where the finite number of converter switch states is exploited to enable an implementation using an explicit (often exhaustive) search.

Several papers have presented research on the design of the cost function [6], [7] and prediction models that can be used. As model based predictive control for power electronics is a computationally intensive scheme, attention has been paid to the implementation technology and methodology as well [6]. It has been shown that in particular an approach using programmable digital electronics to construct a dedicated prediction calculation core offers an efficient implementation.

MBPC has proven to be a powerful tool to control the switches of a power electronic converter. It is an especially attractive control scheme for inverter topologies with a large number of switching states where conventional PWM or SVM modulation scheme's are not obvious. It is however more than a modulation scheme. Besides creating a desired output voltage, MBPC can directly control the load of the inverter. For example in [8], [9] predictive control is used to directly control the torque and flux of an induction machine. Another favorable property is that the control scheme can handle constraints in a natural way. Where other schemes often use hard limitations, MBPC can calculate the optimal switching states, taking the constraints into account [1]. Finally MBPC is typically used to control MIMO systems. Therefore it can also be used to simultaneously control internal inverter states like capacitor or neutral point potentials. Well known examples for which MBPC has proven to be useful are the flying-capacitor inverter [6], [10], the (active) neutral-point-clamped inverter [11]–[13], the cascaded H-brigde inverter [14], [15] and matrix converter [16].

However, as these inverter topologies have a large number of switches, the possibility of a failure of a power electronic switch or driving circuit rises, and with it the need for a fast and reliable fault detection algorithm. Statistical data show that around 38% of the faults in industrial induction machine drives are due to failure of the power electronics [17], whereas 34% of the failure of such power devices is caused by semiconductor component and soldering faults [18]. Literature proposes different open-switch fault detection strategies comparing current or voltage measurements to theoretically expected values as summarized in [17]. In [19] an FPGA was used for real-time fault detection, but the method is based on a voltage-criterion, requiring extra voltage measurements. In [20] open switch faults are detected by looking at the change of the Park's current vector phase. When this value drops under a certain threshold, a fault is assumed. The faulty switch is localized by looking at the polarity of the currents over one fundamental period. This however means that it takes at least

Manuscript received January 13, 2014; revised April 29, 2014 and June 23, 2014; accepted July 21, 2014.

The research was financed by the Research Foundation-Flanders (FWO) project G.0083.13. J. Druant was awarded a Ph.D. Fellowship from the FWO in 2014 (email: Joachim.Druant@ugent.be). J. Druant, F. De Belie, P. Sergeant and J. Melkebeek are with the Electrical Energy Laboratory of the department of Electrical Energy, Systems & Automation of Ghent University, B-9000 Ghent, Belgium. P. Sergeant is also with the Electrical Energy Research Group of the Department of Industrial Technology and Construction, Ghent University, B-9000 Gent, Belgium. (email: Peter.Sergeant@ugent.be). T. Vyncke was with the Department of Electrical Energy, Systems and Automation, Electrical Energy Laboratory, Ghent University, 9000 Gent, Belgium. He is now with Dana Belgium NV, Ten Briele 3, 8200 Brugge, Belgium A Subsidiary of Dana Holding Corporation, Toledo, Ohio, USA.(thomas.vyncke@dana.com, +32486479569) This research has also been carried out in the frame of Interuniversity Attraction Poles Programme initiated by the Belgian Science Policy Office (IAP-VII-02).

one period before the fault can be localized.

The fault detection method introduced here, and presented in a previous paper [21], is an extension of the MBPC algorithm controlling the switches of the converter, so no extra sensors or hardware are needed. It has the additional advantage that measurements and calculations that are already being performed for the control can be recuperated for the fault detection algorithm. The use of an FPGA (field programmable gate array) makes it possible to perform additional calculations used for the online detection in parallel with the control algorithm, so the overall calculation time does not increase. The method can detect open-switch faults in the inverter, and can assign the switch that is causing the fault. In [21] only the basic ideas were given and tested on a linear resistive inductive load used as a simple and comprehensive reference frame. In this paper the detection method is applied to an induction motor, an industrial more relevant application. Although an induction motor is far more difficult to model, the experimental results show fast and reliable fault detection.

This method can be expanded to include a switch-over to fault tolerant operation. The concept of fault tolerant drive system is defined in [22] as the ability to continue to operate in a satisfactory manner after sustaining a fault. In this definition the term ”satisfactory” implies a minimum level of performance after the fault has occurred and thus depends on the system requirements. It allows the drive to continue operation till next maintenance or till a safe stop is possible. Multilevel inverters have the advantage of inherent redundancy that can be exploited for post-fault operation [23], [24]. In [25]–[27] the fault tolerant behaviour of multiphase drives are investigated under an open-phase fault. In [28] also inter-turn and short circuit faults are investigated. Multiphase drives have the advantage that no extra hardware is required to achieve fault tolerant operation. In [25] a predictive controller selects the optimal switching state taking into account the loss of one phase. In most cases however where conventional three-phase drives are used, extra hardware is added to be able to switch to a fault tolerant mode after fault detection. The extra hardware added depends on the type of faults the inverter has to be able to face. This can be a redundant inverter leg [29], but also other more cost-effective topologies are possible depending on the considered faults as in [30]. In this paper, the switch redundant topology of [31] is used, since only single-switch open circuit faults are considered. The authors suggest to decouple the leg containing the faulty switch, and to connect the affected phase to the midpoint of the DC-bus. Automatic controller reconfiguration for subsequent post-fault operation is done by adapting the model in the MBPC controller to the altered inverter configuration. This paper gives measurement results of this kind of reconfiguration.

II. Model Based Predictive Control for Flying-Capacitor Multilevel Inverters

MBPC is an attractive tool to control flying-capacitor multilevel inverters for several reasons. Multilevel inverters typically have a large number of switches to be controlled. These switches have to be controlled in such a way that not only the control objectives are attained, but also that the inverter capacitors remain balanced. MBPC can handle this multiple-input-multiple-output control problem by combining the variables to be controlled in an appropriate cost function. The strategy cannot be seen as just a replacement of existing control schemes like PWM or SVM. The objective of these schemes is to track a desired output voltage or current. The reference values to be tracked are then delivered by another control scheme like for instance a field oriented control scheme for torque and flux control on electric machines. Instead, MBPC directly controls the control variables as will be shown later. The output voltage or current of the inverter is not a goal on itself but a means to reach the control objectives. Another reason to use MBPC is that it can handle restrictions in a natural way. Where other schemes often use hard limitations, MBPC minimizes a cost function taking into account the restrictions.

A. Three level flying-capacitor inverter

The topology of a three level flying-capacitor inverter can be seen in Fig. 1. A flying-capacitor inverter is a kind of multilevel topology meaning that the device can, other than the positive and negative half bus-bar voltage, synthesize several output voltages with respect to the neutral point of the DC-bus. Due to the smaller difference in discrete voltage levels, a voltage wave form with much lower ripple can be achieved. Also much higher voltages can be obtained using conventional switches. This is because several switches are used in series, each of them switching a part of the overall voltage.

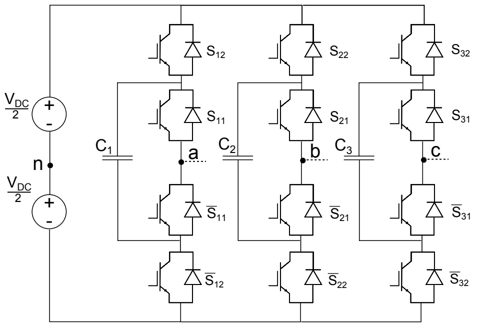


Fig. 1: Topology of a three level flying-capacitor inverter

As can be seen in Fig. 1, this inverter uses two pairs of complementary controlled switches per phase: $(S_{x1}, \bar{S_{x1}})$ and $(S_{x2}, \bar{S_{x2}})$, with $x \in \{1, 2, 3\}$. One extra voltage level can be obtained by controlling the switches in such a way that the flying-capacitor is connected in series with the DC-voltage source. The voltage over the capacitor is balanced to half of the dc-bus voltage. This way an output voltage of zero volts with respect to the neutral point of the DC-bus is obtained. This is the case when $[S_{x1}, S_{x2}] = [1, 0]$ and when $[S_{x1}, S_{x2}] = [0, 1]$. This extra voltage level can be used to synthesize the desired output waveform. The inverter also has redundant switching states which makes it easier to balance the capacitor voltages.

In general, the output voltage is a linear combination of the bus-voltage and the capacitor voltage in that phase:

$$v_{xn} = S_{DC}\frac{V_{DC}}{2} + S_{vc}v_{xc} \qquad x \in \{a,b,c\} \tag{1}$$

V_{DC} is the DC-voltage and v_{xc} is the voltage over the capacitor in phase x.

The variables S_{DC} and S_{vc} are a function of the switching state:

$$S_{DC} = 2S_{x2} - 1 \tag{2}$$

$$S_{vc} = S_{x1} - S_{x2} \tag{3}$$

B. Finite-Set Model Based Predictive Control

Finite-Set Model Based Predictive Control is a specific type of MBPC that can be used to control several outputs. In the case of current tracking control this is the output current and the voltage over the capacitors of the inverter. In the case of predictive torque control (PTC) the output variables are the torque of the motor, the stator flux and the voltage over the capacitors of the inverter. This control algorithm consists of three important steps.

1) ESTIMATION: based on measurements of phase currents, bus voltage and capacitor voltages at the current update instant k an estimation is made for the state at the next update instant $k+1$. To this end the optimal switching state, calculated in the previous time step, is considered. This step compensates for the finite time the controller needs to execute the algorithm.
2) PREDICTION: starting from the estimated values at $k+1$, the output at update-instant $k+2$ is calculated for every possible switching state.
3) OPTIMISATION: for each switching state a cost function is calculated. In this function the relative weight of the different control variables can be chosen, depending on the form of the function. The state with minimum cost is actually applied at time instant $k+1$. All other calculations are removed and the algorithm is iterated.

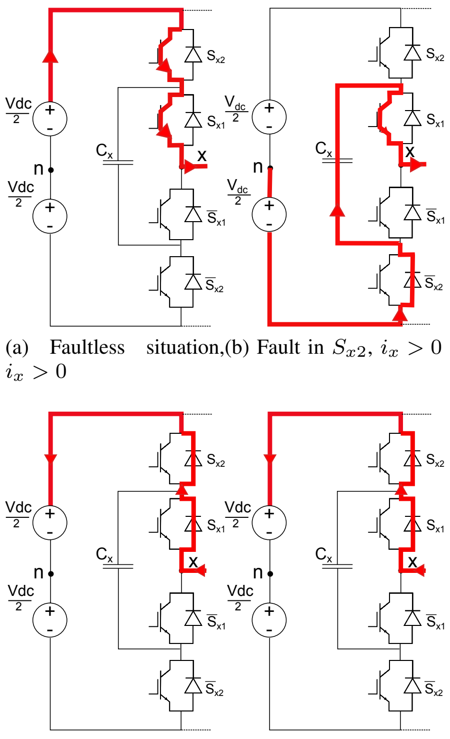
III. Analysis of Open-Switch Inverter Faults in Flying-Capacitor Converters

A. Types of inverter faults and consequences

Power switch failures can be divided into open-switch and closed-switch faults [20]. In this paper a method is presented to detect open-switch inverter faults in a flying-capacitor inverter. An open-switch fault is a situation where a switch does not conduct when a closing signal from the controller is given. This can be due to a fault in the switch itself or a fault in the gate drivers. When this happens, the system model in the controller does not reflect the actual behavior anymore. The estimation and prediction will be wrong and the control will be lost. Note that closed-switch faults are not considered here. In this case the faulted switch does not cease to conduct. This can cause shoot-through and the DC-voltage source to short-circuit. When this happens fuses will decouple the faulty leg from the inverter, reducing the situation to be detected again to an open-switch fault.

B. Modeling of faulty topology

First a model of the system with a fault in each of the switches needs to be derived. To be able to calculate the current through the load in fault condition, the voltage over the load should known. For a three level flying-capacitor inverter the output voltage depends on the switching state, the faulty switch and the sign of the current as will be shown here. In general, the output voltage is a linear combination of the bus voltage and the capacitor voltage in that phase, as shown in equation (1). In a fault situation however, the formulas (2) and (3) are not valid anymore. Instead, the variables S_{DC} and S_{vc} must be determined for each of the twelve fault situations. In each fault situation one of the switches is considered defect. Since the output voltage does not depend on the exact values of the current, values for these variables can be tabulated off-line and can then be used in the detection algorithm. To show how the values of S_{DC} and S_{vc} are obtained, an example is shown here. Suppose the switch S_{x2} of the phase x is defect, the current is positive and that the controller orders both S_{x1} and S_{x2} to close.



(a) Faultless situation,(b) Fault in S_{x2}, $i_x > 0$
$i_x > 0$

(c) Faultless situation,(d) Fault in S_{x2}, $i_x < 0$
$i_x < 0$

Fig. 2: Modeling of faulty topology

The faultless situation is shown in Fig. 2a, where the path of the current is drawn. The output voltage is $+\frac{V_{DC}}{2}$. When a fault occurred in switch Sx2 or its driver, Sx2 will not conduct as demanded by the controller. However, because of the inductive load, the current will continue to flow. Here the freewheel diodes of the switches come into play. As the diode of $\bar{S}_{x1}$ is backward polarized due to the capacitor, the currents flows as in Fig. 2b. Now the output voltage of phase x is $-\frac{V_{DC}}{2}+v_{xc}$, or

$$[S_{DC}, S_{vc}] = [-1, 1] \tag{4}$$

In the other case, when the current was negative, the current flows as in Fig. 2c for the healthy situation. In case of a fault in S_{x2} the current flows as in 2d. In this case there is no difference with the faultless situation. The output voltage of phase x is $\frac{V_{DC}}{2}$, en thus

$$[S_{DC}, S_{vc}] = [1, 0] \tag{5}$$

Finally, if the current crosses zero during the considered update period, it will remain zero since the corresponding freewheel diode will cease conducting.

The fault situations are summarized in table I.

TABLE I: Fault situations (adjacent switch = switch on the same side of DC bus)

Faulty switch	i_x	adjacent switch	$[S_{DC}, S_{vc}]$
S_{x1}	>0	1	[1 ,−1]
		0	[−1 , 0]
	<0	1	[1 , 0]
		0	[−1 , 1]
$\bar{S}_{x1}$	>0	1	[−1 ,0]
		0	[1 , −1]
	<0	1	[−1 , 1]
		0	[1 , 0]
S_{x2}	>0	1	[−1 ,1]
		0	[−1 , 0]
	<0	1	[1 , 0]
		0	[1 , −1]
$\bar{S}_{x2}$	>0	1	[−1 ,0]
		0	[−1 , 1]
	<0	1	[1 , −1]
		0	[1 , 0]

IV. Estimation-based Fault Detection

The proposed fault detection algorithm is based on the comparison between theoretically expected currents and the corresponding measured currents on that time instant. A faulty switch will cause the current path in the inverter to change. This will result in a different voltage applied to the terminals of the load, and thus a different phase current. The detection algorithm calculates what this current would be for every possible open-switch fault. Since there are 12 switches in a three-level flying-capacitor inverter, this results in 12 calculated current vectors. Each current vector thus contains the currents in the three phases that would theoretically occur on the next sampling instant if the corresponding switch would be defect. A 13^{th} current vector contains the estimated phase currents for the faultless situation. This vector is also needed for the control, and thus is already available. This is shown in Fig. 3. In the dotted line the normal FS-MBPC algorithm is depicted. Starting from a measurement on update instant k, an estimation of the state on $k+1$ is made. Starting from this estimation, the state on $k+2$ is predicted for all possible switching states. This creates a finite set of possible future states, from which the optimal state is selected using a cost function. The solid line represents the extension of the algorithm for fault detection. The controller looks at every switch separately and assumes it defect. Then using the measurement of the voltage over the capacitors, the dc-voltage and the sign of the current on update instant k, the phase currents on $k+1$ are estimated for every faulty switch. To this end the table I of output voltages for the different fault situations is used. This results in the 13 current vectors. Twelve vectors contain the three phase currents when one of the twelve switches is defect. The 13^{th} vector contains the current in the three phases for the faultless situation.

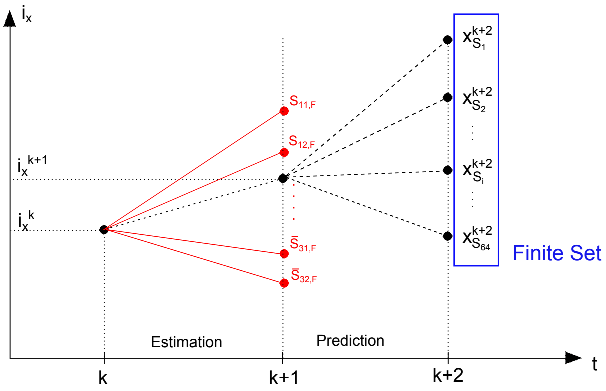


Fig. 3: Principle of fault detection as an extension of the estimation step of the MBPC-algorithm

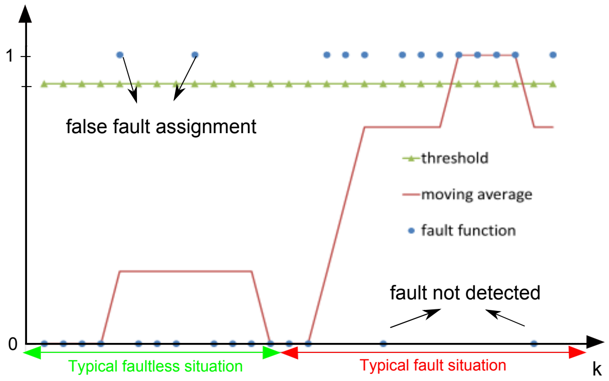
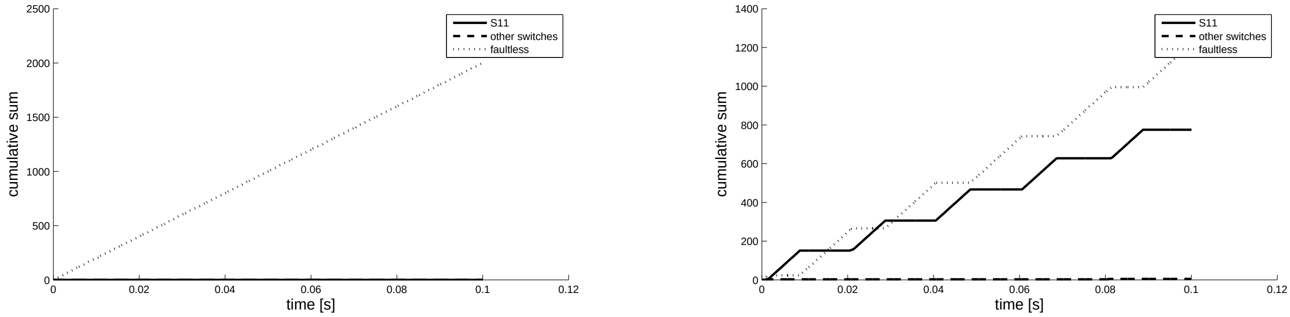


Fig. 4: Principle of fault conclusion. When the moving average of the fault function of a switch exceeds the threshold value, a fault in this switch has occurred.

On update instant $k+1$ a new measurement of the currents comes available. Now the 13 vectors are compared with this measured current vector. This is the minimization step. If the current vector containing the estimated currents for the faultless situation is closest to the measured vector on $k+1$ it



(a) Simulation of the cumulative sum of the fault function for the different switches when no fault is present.

(b) Simulation of the cumulative sum of the fault function for the different switches, a fault in S_{11} is simulated.

Fig. 5: Fault detection method: simulation results

is assumed that none of the switches is defect. If however one of the twelve fault-vectors is closest to the measured vector on $k+1$, it is possible that the corresponding switch is defect. A fault is not immediately concluded, because measurement noise and modeling uncertainties could cause a false conclusion. Instead a fault function is associated to each switch. This function has the value 1 on time instant k if the corresponding switch was chosen in the minimization step on update instant k. Otherwise this function is zero. On each instant a moving average of this function is calculated for each switch to filter out the false fault assignments, as can be seen in Fig. 4. Also in some cases the fault cannot immediately be detected since there is no difference between the faulty and faultless situation, as seen in section III-B. When the moving average exceeds a certain threshold, a fault in the corresponding switch is assumed.

V. Fault detection on a three level flying-capacitor inverter with linear resistive inductive load

To test the detection algorithm, the simulations and experimental verifications were first carried out on an RL-load. This load has the advantage that it is straightforward to model so the focus can be laid on the detection of faults. The load has a resistance of $R = 4.5\Omega$, and an inductance of $L = 0.0145H$. The three level flying-capacitor inverter has capacitors of $C = 110\mu F$. This value is a trade off between controllability of the voltage and cost/weight. The inverter is fed with a DC-voltage of $100V$, the rated voltage of the three level flying-capacitor inverter.

A. Simulation Results

Simulations, carried out in a Matlab-Simulink environment, are performed to check whether the detection algorithm selects the right switch. To do this the cumulative sum of the faultfunction is plotted in Fig. 5. This function tells how often a certain switch has been chosen by the minimization step of the fault detection algorithm. To test the algorithm this cumulative sum which corresponds with the twelve switches together with the faultless situation, was simulated for different fault situations. First a simulation was made with no faulty switches. The result is given in Fig. 5a. As expected, the faultless situation was chosen every time.

Next, a fault in switch S_{11} was simulated. The result is shown in Fig. 5b. It is clear that the detection algorithm detects the fault as the cumulative sum of S_{11} rises fast. Also the faultless situation is selected a lot of times. This is because S_{11} is not used all the time. Wen S_{11} is not used, there is no difference between the faultless and the fault situation.

B. Experimental Verification

To verify the results, the detection algorithm was implemented in a Xilinx Virtex-II FPGA, as can be seen in Fig. 6 where the measurement setup is shown. The FPGA was programmed using ISE 10.1 design suite from Xilinx. The used FPGA has 13696 slices of which 32% are used for the current control, and an additional 10% for the fault detection.

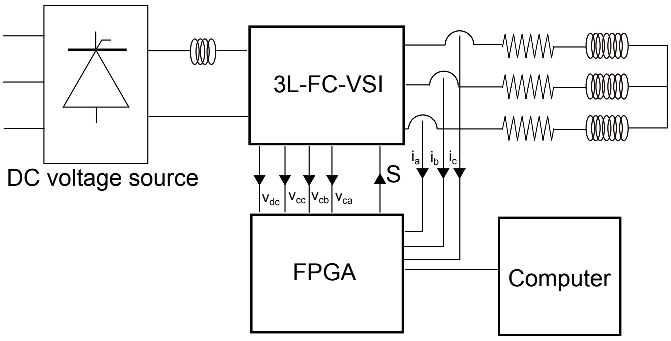


Fig. 6: Measurement setup

To emulate a faulty switch, the switch S_{22} could be shut down. The result is shown in Fig. 7. The current reference wave is plotted together with the actual current. At $t = 0$ ms a fault in switch S_{22} was emulated by sending a constant zero signal to its gate. From this moment, the currents start to deviate from their reference. The moving average of the fault function of this switch starts to rise, and reaches the threshold value. The controller concludes there is a fault in switch S_{22}, and some action can be undertaken as will be shown later. The time between fault introduction and detection is 1.22ms here. This value of course depends on the threshold value. If this value is chosen lower, the fault can be detected earlier. On the other hand the risk that a false fault conclusion is made rises. Here a threshold of 0.60 was chosen from trial and error. This value is far higher than the maximum of the moving average in faultless situation, making the algorithm robust to false alarms. On the other hand the value is lower than the moving average in fault situation making the detection sensitive to faults. The time between fault introduction and detection also depends on the value of the reference signal at the moment of fault introduction. In Fig. 8 for instance, the fault was introduced at negative reference value. The currents keeps following the reference, until the reference becomes positive. This can be explained by looking at the inverter topology. If the current is negative, the switch S_{22} is hardly used because the current flows mainly through $\bar{S}_{21}$ and $\bar{S}_{22}$ to the negative terminal of the DC-bus. The faulty switch is only used to balance the capacitor in phase b. This is why it takes about 4ms before the fault is detected. This is not really a problem since most of this time the current tracks the reference.

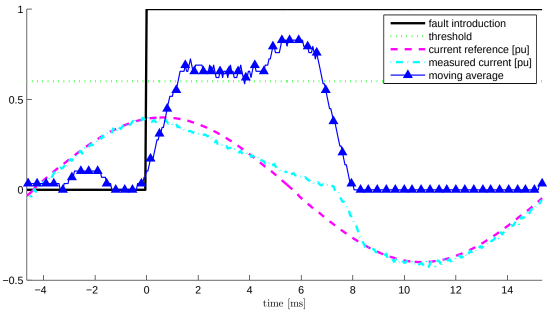


Fig. 7: Experimental verification, fault on switch S_{22}.

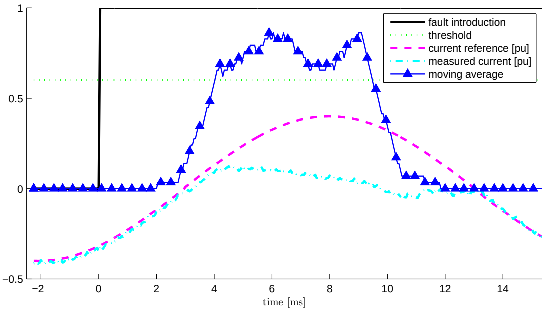
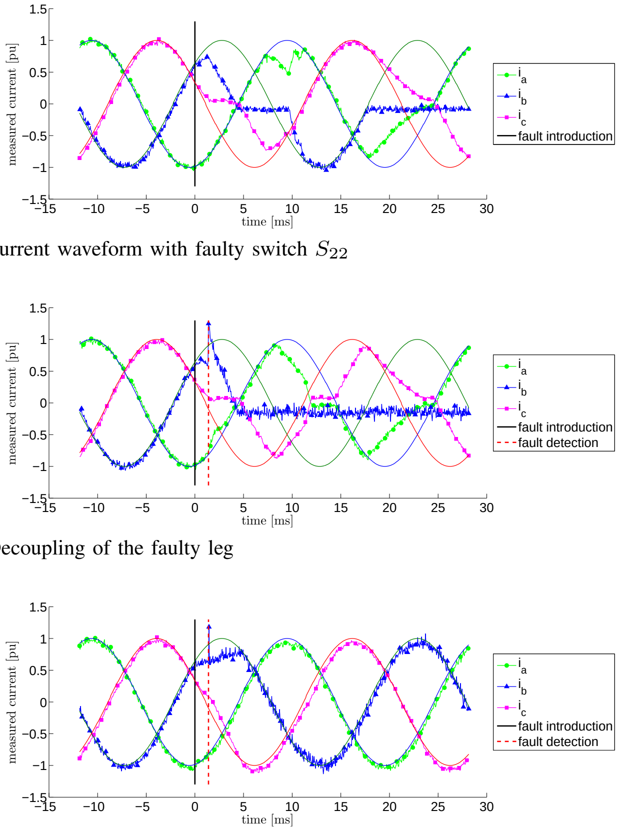


Fig. 8: Experimental verification, fault on switch S_{22}. Fault introduction at different angle with respect to reference current.

C. Reconfiguration after Fault Detection

After a fault in a switch has been detected, some actions can be undertaken to minimize the effect of the fault on the output current of the inverter. In this paper a proposed method [31] in which the affected phase is connected with the neutral point of the DC-bus is investigated. All measured waveforms are plotted in Fig. 9. The measurement setup of Fig. 6 was used (RL-load). Fig. 9a shows the current waveforms when a fault in switch S_{22} occurs and no further action is undertaken. From the moment of fault introduction all currents start to deviate from the reference values. Since phase b cannot be connected to the positive DC-bar due to the fault in S_{22}, the current in that phase always remains negative or zero. In Fig. 9b the affected phase is, after the fault has been detected, fully decoupled from the inverter. Obviously the current in this phase always remains zero. The controller does not manage to control the two other currents either. Finally, Fig. 9c shows what happens when the affected phase is connected to the mid-point of the DC-bus using a bidirectional switch. The fault is detected after 1.4 ms. From that moment the current of phase b flows through the bidirectional switch. This information is passed to the controller, and the model of the inverter is altered. The phase voltage of b is always set to zero in the model independent of the switching state. Using this new model the currents can be controlled in an acceptable way. This could be a temporary solution until the faulty switch is replaced.



(a) Current waveform with faulty switch S_{22}

(b) Decoupling of the faulty leg

(c) Connecting the affected phase with the neutral point of the DC-bus and adapting the model

Fig. 9: Reconfiguration after fault detection, measurement data.

VI. Fault detection on a three level flying-capacitor inverter controlling an induction machine

Now that it is shown that the principle of fault detection makes sense, an industrially more relevant application is considered. In the section below a three level flying-capacitor VSI is used to drive an induction motor. MBPC is used to directly control the torque and flux of the motor [8], [9] along with the voltage over the capacitors of the inverter. The inverter is fed with a DC-voltage of $100V$, the rated voltage of the three level flying-capacitor inverter. A DC-machine is connected to the 380V induction machine to serve as load in order to keep the speed and thus the back-emf low so that no overmodulation takes place. The speed of the machine is in all simulations and experimental verifications kept at 150 r.p.m. The considered induction motor is a Leroy Somer motor of the type LS112MT of which the parameters are given in table II.

TABLE II: Motor parameters converted to stator.

Leroy Somer Induction Motor	
Type LS112MT	Nr. 061300JM017
$R_s = 1.23\Omega$	$R_r = 1.05\Omega$
$L_m = 0.212H$	$L_s = L_r = 0.220H$
$P_{nom} = 4kW$	$N_{nom} = 2815rpm$

A. Simulation results

First, simulations were performed to see whether the detection algorithm is robust enough to cope with model imperfections. Again the cumulative sum of the fault function is plotted. A fault in switch S_{11} was simulated at $t = 0.5$ s. The result is shown in Fig. 10. It is clear that the detection algorithm detects the fault as the cumulative sum of S_{11} rises fast. Also the faultless situation is selected a lot of times. This is because S_{11} is not used constantly. When S_{11} is not used, there is no difference between the faultless and the fault situation.

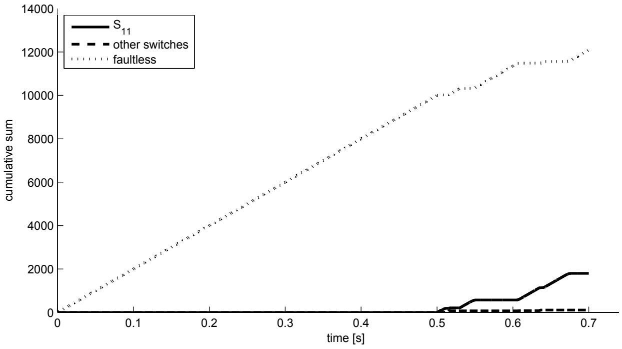


Fig. 10: Simulation of the cumulative sum of the fault function for the different switches, a fault in S_{11} is simulated at $t = 0.5$ s.

B. Experimental Verification

To verify the results, the detection algorithm is implemented in a Xilinx Virtex-II FPGA along with the PTC algorithm to control the inverter. The measurement setup is the same as in Fig. 6 but with an induction machine as load. The experimental test rig is shown in Fig. 11.

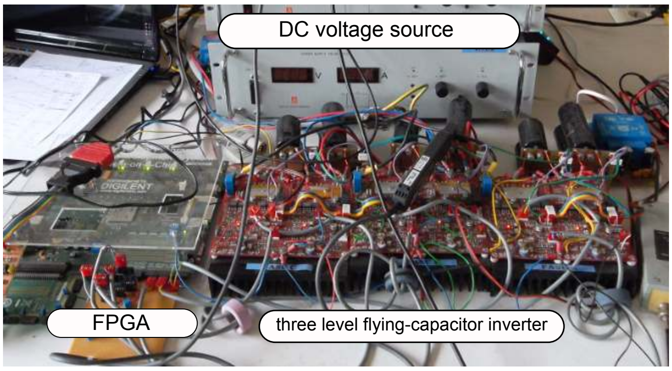
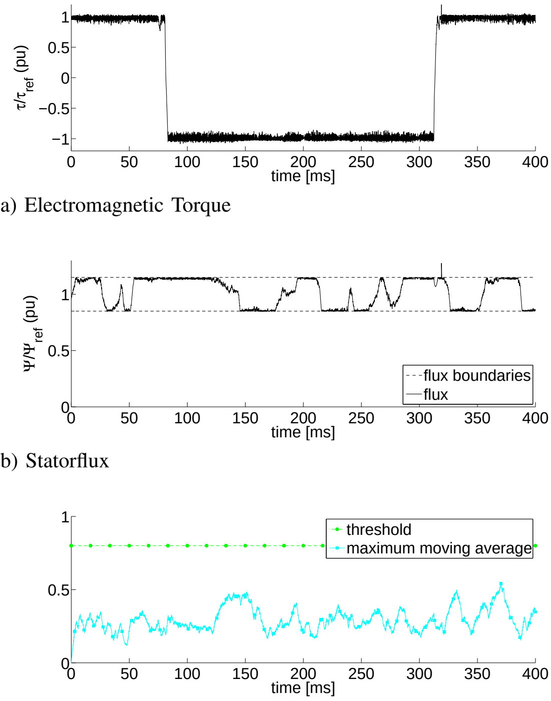


Fig. 11: Measurement setup

1) Pre-fault behaviour: A very important feature of a fault detection algorithm is its ability to avoid false fault conclusions. To this end the detection was tested for a negative and positive step in the torque reference. The result can be seen in Fig. 12 where the torque, the stator flux and the fault detection are shown. The predictive controller tracks the steps in the torque accurately while maintaining the desired flux level between its boundaries. In figure 12c the maximum value at each instant of all filtered fault functions are shown. From the figure it is clear that no false fault conclusions are made, even when sharp torque transients are imposed. The maximum value is 0.55, while the threshold is chosen to be 0.80. These values might change when another inverter or machine is used. So the moving average and the threshold is something to be tuned, which is a disadvantage of this method. However when the right tuning variables are obtained, a robust fault detection results.

2) Fault detection: The next feature to be tested is the detection algorithm's sensitivity to a fault. To emulate a faulty switch, the switch S_{11} can be shut down. The result is shown in Fig. 13. At $t = 0$ ms a fault in switch S_{11} was emulated by sending a constant zero signal to its gate. The moving average of the fault function of this switch starts to rise, and reaches the threshold value. From this moment the controller concludes there is a fault in switch S_{11}. The time between fault introduction and detection is 49 ms here. The controller however can keep track after the fault introduction until the torque drops after 45ms, as can be seen in Fig. 13a. The reason for this is that the controller does not need the faulty switch a lot during this period to regulate the torque and the flux. So the time between fault-introduction and loss of control is only a few milliseconds. This value of course depends on the threshold value.

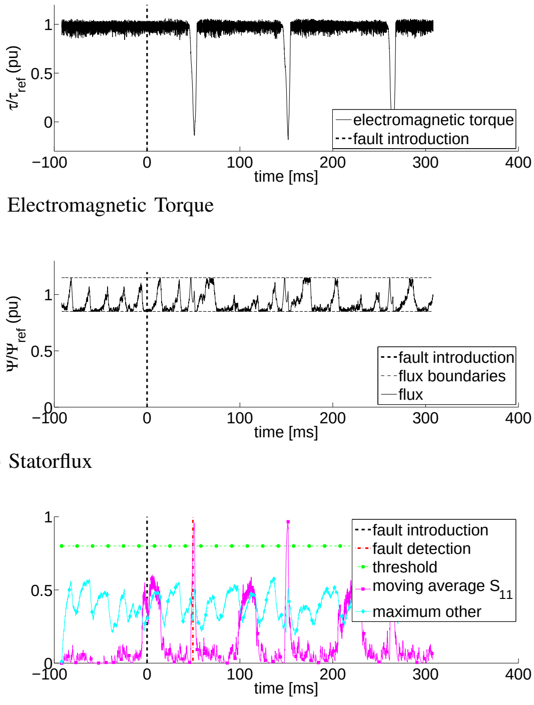


(a) Electromagnetic Torque

(b) Statorflux

(c) Maximum of all filtered fault functions.

Fig. 12: Predictive torque control, measurement data. The controller tracks the torque reference steps accurately while maintaining the statorflux. No false fault conclusions are made.

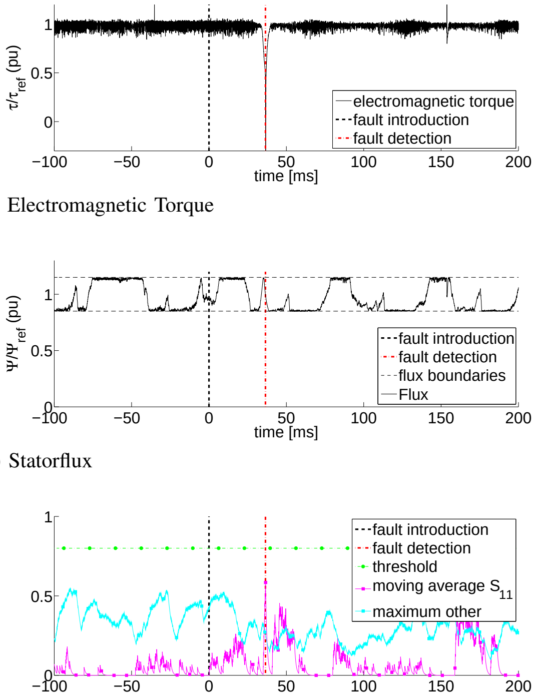


(a) Electromagnetic Torque

(b) Statorflux

(c) Fault detection.

Fig. 13: Estimation based fault detection, measurement data. The fault in S_{11} is detected as the corresponding moving average exceeds the threshold value.

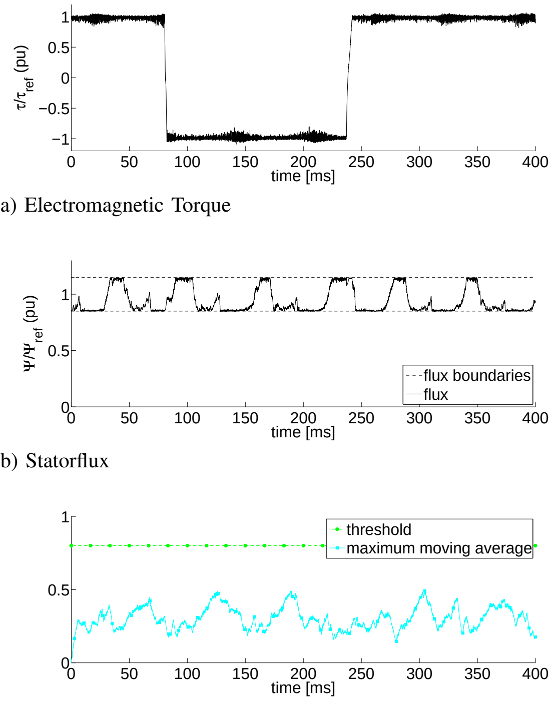


(a) Electromagnetic Torque

(b) Statorflux

(c) Fault detection.

Fig. 14: Reconfiguration after fault detection, measurement data. After a fault has been detected, the controller switches to fault tolerant mode while the affected phase is connected to the midpoint of the DC-bus.



(a) Electromagnetic Torque

(b) Statorflux

(c) Fault detection.

Fig. 15: Post fault operation, measurement data. Even with two phases, the predictive controller can track the steps in the torque reference and maintain the flux level.

If the threshold is chosen lower, the fault can be detected earlier. On the other hand the risk that a false fault conclusion is made rises. Here a threshold of 0.80 was chosen which is larger than in the case of an RL-load in order to compensate for model imperfections. Also the maximum moving average of the fault function of all other switches is plotted in Fig. 13c to show that these functions do not reach the threshold when a fault occurs in S_{11}.

3) Reconfiguration after fault detection: As explained in V-C, some action can be undertaken to minimize the effect of the fault on the torque and flux control. Fig. 14 shows what happens when the phase containing the fault is decoupled from the inverter and connected to the midpoint of the DC-bus. Fig. 14a shows that the torque shows a small dip caused by the delay needed to detect the fault, but that the torque can be controlled accurately in fault tolerant mode. Note that in fault tolerant mode the fault function of S_{11} does not cross the threshold anymore since the model of the inverter is altered to the new configuration.

4) Post-fault behaviour: To check the torque response in fault tolerant mode, some torque transients were imposed in figure 15. Even with only two inverter legs, predictive torque control is able to accurately control the torque and the flux. The machine can thus continue to operate in this mode until a safe stop is possible. No extra pressure is put on the other working switches. The detection algorithm is now able to detect other faults in the remaining switches. After a second fault is detected, only one phase remains. In this case post fault operation is not possible anymore.

VII. Conclusion

In this paper a model-based fault detection algorithm is proposed. The method is an extension of the estimation step in the FS-MBPC algorithm. The fault in one of the switches is detected by comparing the real measured current with the currents theoretically expected. It is concluded that a fault in a power switch and the switch responsible for it can be detected in a fast and reliable way both on a linear inductive resistive load and on an induction machine. On the contrary to other fault detection methods in the literature, no additional hardware is needed. Furthermore, the method is not only cost-effective but can also systematically be expanded to topologies with a higher number of switches. When a fault is detected, the faulty leg can be decoupled and the affected phase is connected to the midpoint of the DC-bus. Experimental data shows that the torque can still be controlled in an affordable way.

References

[1] S. Kouro, P. Cortés, R. Vargas, U. Ammann, and J. Rodríguez, “Model predictive control-a simple and powerful method to control power converters,” *IEEE Trans. Ind. Electron.*, vol. 56, no. 6, pp. 1826–1838, Jun. 2009.

[2] Y. Zhang, W. Xie, Z. Li, and Y. Zhang, “Model predictive direct power control of a pwm rectifier with duty cycle optimization,” *IEEE Trans. Power Electron.*, vol. 28, no. 11, pp. 5343–5351, Nov 2013.

[3] A. Linder, R. Kanchan, R. Kennel, and P. Stolze, *Model-Based predictive control of electric drives*. Göttingen: Cuvillier Verlag, 2010.

[4] C. Hackl, F. Larcher, A. Dotlinger, and R. Kennel, “Is multiple-objective model-predictive control optimal?” in *IEEE International Symposium on Sensorless Control for Electrical Drives and Predictive Control of Electrical Drives and Power Electronics (SLED/PRECEDE)*, Oct 2013, pp. 1–8.

[5] J. Rodriguez, M. Kazmierkowski, J. Espinoza, P. Zanchetta, H. Abu-Rub, H. Young, and C. Rojas, “State of the art of finite control set model predictive control in power electronics,” *IEEE Trans. Ind. Informatics*, vol. 9, no. 2, pp. 1003–1016, May 2013.

[6] T. J. Vyncke, S. Thielemans, and J. A. Melkebeek, “Finite-set model based predictive control for flying-capacitor inverters: cost function design and efficient FPGA implementation,” *IEEE Trans. Ind. Informatics*, vol. 9, no. 2, pp. 113–1121, May 2013.

[7] S. Thielemans, T. Vyncke, and J. Melkebeek, “Weight factor selection for model-based predictive control of a four-level flying-capacitor inverter,” *Power Electronics, IET*, vol. 5, no. 3, pp. 323–333, 2012.

[8] C. Rojas, J. Rodriguez, F. Villarroel, J. Espinoza, C. Silva, and M. Trincado, “Predictive torque and flux control without weighting factors,” *IEEE Trans. Ind. Electron.*, vol. 60, no. 2, pp. 681–690, 2013.

[9] T. Geyer, G. Papafotiou, and M. Morari, “Model predictive direct torque control part i: Concept, algorithm, and analysis,” *IEEE Trans. Ind. Electron.*, vol. 56, no. 6, pp. 1894–1905, 2009.

[10] A. Shukla, A. Ghosh, and A. Joshi, “Improved multilevel hysteresis current regulation and capacitor voltage balancing schemes for flying capacitor multilevel inverter,” *IEEE Trans. Power Electron.*, vol. 23, no. 2, pp. 518–529, March 2008.

[11] J. Barros, J. Silva, and E. Jesus, “Fast-predictive optimal control of npc multilevel converters,” *IEEE Trans. Ind. Electron.*, vol. 60, no. 2, pp. 619–627, 2013.

[12] K. Wang, Z. Zheng, Y. Li, K. Liu, and J. Shang, “Neutral-point potential balancing of a five-level active neutral-point-clamped inverter,” *IEEE Trans. Ind. Electron.*, vol. 60, no. 5, pp. 1907–1918, 2013.

[13] F. Kieferndorf, P. Karamanakos, P. Bader, N. Oikonomou, and T. Geyer, “Model predictive control of the internal voltages of a five-level active neutral point clamped converter,” in *IEEE Energy Conversion Congress and Exposition (ECCE)*, 2012, pp. 1676–1683.

[14] K. Corzine and Y. Familiant, “A new cascaded multilevel h-bridge drive,” *IEEE Trans. Power Electron.*, vol. 17, no. 1, pp. 125–131, Jan 2002.

[15] P. Cortes, A. Wilson, S. Kouro, J. Rodriguez, and H. Abu-Rub, “Model predictive control of multilevel cascaded h-bridge inverters,” *IEEE Trans. Ind. Electron.*, vol. 57, no. 8, pp. 2691–2699, 2010.

[16] M. Rivera, C. Rojas, J. Rodriguez, P. Wheeler, B. Wu, and J. Espinoza, “Predictive current control with input filter resonance mitigation for a direct matrix converter,” *IEEE Trans. Power Electron.*, vol. 26, pp. 2794–2803, Oct. 2011.

[17] B. Lu and S. Sharma, “A literature review of igbt fault diagnostic and protection methods for power inverters,” *IEEE Trans. Ind. Applications*, vol. 45, no. 5, pp. 1770–1777, 2009.

[18] S. Yang, D. Xiang, A. Bryant, P. Mawby, L. Ran, and P. Tavner, “Condition monitoring for device reliability in power electronic converters: A review,” *IEEE Trans. Power Electron.*, vol. 25, no. 11, pp. 2734–2752, Nov 2010.

[19] M. Shahbazi, P. Poure, S. Saadate, and M. Zolghadri, “Fpga-based fast detection with reduced sensor count for a fault-tolerant three-phase converter,” *IEEE Trans. Ind. Informatics*, vol. 9, no. 3, pp. 1343–1350, Aug 2013.

[20] N. Freire, J. Estima, and A. Marques Cardoso, “Open-circuit fault diagnosis in pmsg drives for wind turbine applications,” *IEEE Trans. Ind. Electron.*, vol. 60, no. 9, pp. 3957–3967, Sept 2013.

[21] J. Druant, T. Vyncke, and J. Melkebeek, “Adding inverter fault detection to model-based predictive control for flying-capacitor inverters,” in *IEEE International Symposium on Sensorless Control for Electrical Drives and Predictive Control of Electrical Drives and Power Electronics (SLED/PRECEDE)*, Oct 2013, pp. 1–5.

[22] B. Welchko, T. Lipo, T. Jahns, and S. Schulz, “Fault tolerant three-phase ac motor drive topologies: a comparison of features, cost, and limitations,” *IEEE Trans. Power Electron.*, vol. 19, no. 4, pp. 1108–1116, 2004.

[23] Y. Song and B. Wang, "Survey on reliability of power electronic systems," *IEEE Trans. Power Electron.*, vol. 28, no. 1, pp. 591–604, Jan 2013.

[24] J. Li, A. Huang, Z. Liang, and S. Bhattacharya, "Analysis and design of active npc (anpc) inverters for fault-tolerant operation of high-power electrical drives," *IEEE Trans. Power Electron.*, vol. 27, no. 2, pp. 519–533, 2012.

[25] H. Guzman, M. Duran, F. Barrero, B. Bogado, and S. Toral, "Speed control of five-phase induction motors with integrated open-phase fault operation using model-based predictive current control techniques," *IEEE Trans. Ind. Electron.*, vol. 61, no. 9, pp. 4474–4484, Sept 2014.

[26] H. Che, M. Duran, E. Levi, M. Jones, H. Ping, and N. Abd Rahim, "Post-fault operation of an asymmetrical six-phase induction machine with single and two isolated neutral points," *IEEE Trans. Power Electron.*, vol. PP, no. 99, pp. 1–1, 2013.

[27] M. Salehifar, R. Arashloo, J. Moreno-Equilaz, V. Sala, and L. Romeral, "Fault detection and fault tolerant operation of a five phase pm motor drive using adaptive model identification approach," *IEEE Journal of Emerging and Selected Topics in Power Electronics*, vol. 2, no. 2, pp. 212–223, June 2014.

[28] A. Mohammadpour and L. Parsa, "Global fault-tolerant control technique for multi-phase permanent-magnet machines," *IEEE Trans. Ind. Applications*, vol. PP, no. 99, pp. 1–1, 2014.

[29] R. Errabelli and P. Mutschler, "Fault-tolerant voltage source inverter for permanent magnet drives," *IEEE Trans. Power Electron.*, vol. 27, no. 2, pp. 500–508, 2012.

[30] W. Wang, M. Cheng, B. Zhang, Y. Zhu, and S. Ding, "A fault-tolerant permanent-magnet traction module for subway applications," *IEEE Trans. Power Electron.*, vol. 29, no. 4, pp. 1646–1658, 2014.

[31] R. L. de Araujo Ribeiro, C. B. Jacobina, E. R. Cabral da Silva, and A. M. N. Lima, "Fault-tolerant voltage-fed PWM inverter AC motor drive systems," *IEEE Trans. Ind. Electron.*, vol. 51, no. 2, pp. 439–446, Apr. 2004.

Joachim Druant was born in Ieper, Belgium on February 19, 1990. He received the M.Sc. degree in electromechanical engineering in 2013 from Ghent University, Ghent, Belgium. Since then, he has been with the Electrical Energy Laboratory (EELAB), Department of Electrical Energy, Systems and Automation (EESA) of Ghent University and is currently working towards a Ph.D. degree. In 2014, he was awarded a Ph.D. Fellowship from the Research Foundation-Flanders (FWO). His present research interests include digital control of converter-fed electrical machines, fault tolerant control, and modeling and control on electrical variable transmissions.

Thomas Vyncke (S'06-M'12) was born in Oostende, Belgium, in 1982. He received the M.Sc.degree in electromechanical engineering and Ph.D. degree from Ghent University, Ghent, Belgium, in 2005 and 2012, respectively. In his Ph.D. dissertation on predictive controllers for direct torque control of AC machines, he researched FS-MBPC schemes and their implementation in programmable digital hardware. He is currently System Engineer Advanced Powertrain Engineering at Dana Belgium NV. His research interests include power electronics, motor control, advanced control strategies for mechatronic applications, and conceptual design of vehicle drivetrain solutions.

Frederik De Belie was born in Belgium in 1979. He received the Master degree in electromechanical engineering from Ghent University, Ghent, Belgium, in 2002, and the Ph.D. degree in March 2010. He's currently a post-doctoral assistant in the Electrical Energy, Systems and Automation Department of the Ghent University. His present research interests include modelling theory and control-system theory applied to electrical drives and, in particular, self-sensing control of synchronous machines.

Peter Sergeant received the M.Sc. degree in electromechanical engineering in 2001, and the Ph.D. degree in engineering sciences in 2006, both from Ghent University, Ghent, Belgium. In 2001, he became a researcher at the Electrical Energy Laboratory of Ghent University. He became a postdoctoral researcher at Ghent University in 2006 (postdoctoral fellow of the Research Foundation - Flanders) and at Ghent University College in 2008. Since 2012, he is associate professor at Ghent University. His current research interests include numerical methods in combination with optimization techniques to design nonlinear electromagnetic systems, in particular, electrical machines for sustainable energy applications.

Jan Melkebeek was born in Gent, Belgium on February 20, 1952. He received the 'Ingenieur' degree in electrical and mechanical engineering from the University of Gent, Belgium in 1975, and the degree of Doctor in Applied Sciences from the same university in 1980. In 1986 he obtained the degree of 'Doctor Habilitus' in Electrical and Electronical Power Technology. Since 1987 he is Professor in Electrical Engineering at the Engineering Faculty of the University of Gent. He is the head of the Department of Electrical Power Engineering, Systems and Automation and the director of the Electrical Energy Laboratory (EELAB). Prof. Melkebeek is a fellow of the IEE and a senior member of the IEEE. He served as the president of the IEEE Benelux IAS-PELS joint chapter from 2002 to 2003 and is a long-time member of the IEEE-IAS Electric Machines Committee, the IEEE-IAS Electric Drives Committee and of the IEEE-PES Machine Theory Subcommittee.