Fault Detection, Isolation and Control Reconfiguration of Three-Phase PMSM Drives

Fabien Meinguet*, Xavier Kestelyn§, Eric Semail§ and Johan Gyselinck*
*BEAMS Department, Université Libre de Bruxelles (ULB), Brussels, Belgium
e-mail: Fabien.Meinguet@ulb.ac.be
§L2EP, Arts et Métiers ParisTech, Lille, France

Abstract—This paper deals with on-line software fault detection and isolation method for a drive composed of a four-leg inverter and a three-phase permanent magnet synchronous machine. The considered faults are single-phase open-circuit and current sensor outage. The method is based on the monitoring of the abc currents with phase-locked loops and the ‘CUSUM’ algorithm for the decision system. The impact of the considered faults is examined: first, in case there is no modification of the control and then in case a control reconfiguration is performed taking into account the fault diagnosis. Closed-loop operation is performed before, during and after the fault. Experimental results show that the latter case allows maintaining the drive in safe operation.

I. INTRODUCTION

The need for reliability and continuous operation has lead all over the years to the development of fault-tolerant electrical drives for various industrial purposes and for transport applications [1], [2]. Although permanent magnet synchronous machines (PMSM) are not inherently fault tolerant, especially to short-circuit in the windings, their power density and absence of maintenance make them a good trade-off in applications where weight and volume are particularly important.

Fault detection can be generally performed with hardware devices and/or software methods. Hardware devices have to be used when software methods are unsuitable, i.e. when the characteristic time of the fault is much smaller than the sampling period. An example is a short-circuited IGBT that can be managed with gate drive protection. Hardware devices are generally costly and complex but can provide a fast and reliable detection, while software requires only computation time but can be slower.

Survey papers dedicated to machine faults describe very well the different faults and various software methods to detect them [3]–[5]. Current-based methods are obviously attractive for drives since the currents are measured for the control and protections anyway. Hence, a current measurement-based diagnostic can be implemented without additional cost.

Inverter faults and fault-tolerant three-phase topologies are investigated in [6]. Redundancy is adopted to cope with switch faults and the authors suggest to access the neutral point of the machine or to supply each phase individually to keep the motor in operation after the fault mitigation. Considering PMSM drives, the neutral accessibility may be an issue since most of the commercial PMSMs have no neutral connection. However, fault-tolerance should be taken into account from the beginning when designing a fault-tolerant drive. The neutral connection can cause different problems that can be solved with an appropriate PMSM design. First, the electromotive force (emf) zero-sequence component should be as low as possible. It prevents the flow of a current zero-sequence component that creates torque ripple, unless a complex current control is implemented [7]. Secondly, the zero-sequence inductance can be much lower than the dq-axis inductance, leading to a high current ripple and losses. This can be avoided by choosing a design with no mutual inductance, for instance with a fractional-slot winding and non-overlapping coils [8].

The authors of [9] propose the idea of detecting mechanical sensor faults by comparing the output of the sensor with the estimation of a sensorless algorithm. They also suggest detecting dc-link voltage sensor fault based on a power balance equation. Moreover, a test at standstill assessing the condition of the current sensors is detailed. In case of current sensor fault, observers are used to reconstruct the missing data.

Current sensor fault detection and isolation (FDI) is studied in [10]–[13] with two different approaches: in [10], a model-based fault detection and isolation is performed, whereas [11]–[13] uses a signal-based approach. A shortcoming of these methods is that they all perform the fault detection and isolation with an open-loop control, except in [13] where the fault is detected but not isolated. Generally, model-based methods are sensitive to parameter variations and/or uncertainties, while signal-based methods present a lack of performance with a closed-loop controller.

Beyond the ease of repair brought by the FDI, it makes fault-tolerant control (FTC) possible since the controller knows which component is faulty and can adapt the control strategy accordingly. For example, FTC for induction motor drives is presented in [14], where an architecture that changes the control algorithm in function of the available sensors is presented. A different way to achieve FTC is to take benefit of the additional degrees of freedom in case a fault tolerant topology is used. For example, it is possible to reconfigure the control to drive the machine with two phases and the neutral point [15], [16].

In this paper, we consider a fault-tolerant topology where the inverter has one additional leg. By connecting this leg continuously to the neutral point of the machine, we show how the effect of a fault is reduced. This is particularly important since the software FDI method needs some time to be executed and the system needs then to withstand the fault while waiting for a control reconfiguration.

The proposed FDI method is based on a phase-locked loop (PLL) [12], [17] and the CUSUM algorithm for the decision
By applying the Park transformation defined by:

\[
[X_{dq0}] = [P][X_{abc}] ; [X_{abc}] = [P]^{-1}[X_{dq0}]
\]

\[
[P] = \frac{2}{3}[R][C] ; [P]^{-1} = [C]^t [R]^t
\]

\[
[R] = \begin{bmatrix}
\cos \theta_e & \sin \theta_e & 0 \\
-\sin \theta_e & \cos \theta_e & 0 \\
0 & 0 & 1
\end{bmatrix} ; [C] = \begin{bmatrix}
1 & \frac{-1}{\sqrt{3}} & \frac{-1}{\sqrt{3}} \\
\frac{1}{\sqrt{3}} & 1 & \frac{-1}{\sqrt{3}} \\
\frac{1}{\sqrt{3}} & \frac{-1}{\sqrt{3}} & 1
\end{bmatrix}
\]

where \( \theta_e \) is the electrical position of the rotor. Applying the transformation (3) to the natural system (1) gives:

\[
\begin{bmatrix}
V_d \\
V_q \\
V_0
\end{bmatrix} = R_s \begin{bmatrix}
I_d \\
I_q \\
I_0
\end{bmatrix} + \begin{bmatrix}
0 & 0 & 0 \\
0 & L_{dq} & 0 \\
0 & 0 & L_0
\end{bmatrix} \frac{d}{dt} \begin{bmatrix}
I_d \\
I_q \\
I_0
\end{bmatrix} + \omega_e \begin{bmatrix}
-L_{dq} I_q \\
L_{dq} I_d + e_{M,1} \\
e_{M,3}(\theta_e)
\end{bmatrix}
\]

\[
L_{dq} = L - M ; L_0 = L + 2M \left( -\frac{1}{2} L \leq M \leq 0 \right)
\]

where \( \omega_e \) is the electrical pulsation, \( e_{M,1} \) is the amplitude of the magnet fundamental linkage flux and \( e_{M,3}(\theta_e) \) is the emf third harmonic component for each electrical radian, which a sinusoidal function of \( 3\theta_e \).

The electromagnetic torque expressed in the \( dq0 \) reference frame is equal to:

\[
T_{em} = \frac{3}{2} P (e_{M,1} I_q + e_{M,3}(\theta_e) I_0)
\]

where \( P \) is the number of pole pairs.

Parameters of the machine used for the tests are: \( R_s = 1.39 \Omega \), \( L_{dq} = 11.4 mH \), \( L_0 = 4.9 mH \), \( \psi_{M,1} = 1.05 V s/rad \), \( c_{M,2} = 0.06 \), \( P = 3 \). Ratings are: \( \Omega_m = 2000 rpm, \, P = 2.59 kW, \, I_{nom} = 5.7 A \). As the emf third harmonic component is much smaller than the fundamental component for the considered machine, the zero-sequence component contribution to the electromagnetic torque (8) will be neglected in what follows.
A. Phase-locked loop

Various types of PLL have been proposed in the literature. A PLL expressed in the synchronous reference frame (SRF) has been implemented [17]. An additional adaptive filtering stage has been added at the input of the PLL, as shown in Fig. 4. It consists of a structure called second order generalized integrator-based quadrature signals generator (SOGI-GSQ) [19]. Its role is to eliminate harmonics and to generate a quadrature signal needed by the SRF PLL. The $D$ filter transfer function is equal to:

$$D(s) = \frac{X'(s)}{X(s)} = \frac{k\omega_e s}{s^2 + k\omega_e s + \omega_e^2} \quad (9)$$

where $X'(s)$ is the filter output, $k$ a damping factor, $s$ the Laplace variable and $\omega_e$, the electrical pulsation, is the resonant frequency of the filter.

The quadrature signal generator is a filter of which the transfer function $Q(s)$ is:

$$Q(s) = \frac{qX'(s)}{X(s)} = \frac{k\omega_e^2}{s^2 + k\omega_e s + \omega_e^2} \quad (10)$$

An example is shown in Fig. 5. A fault on the $a$-phase has been performed at the time $t = 0.84s$. When the fault occurs, the signal $\omega_{PLL,I_a}$ drops, but not immediately to zero, while the value of the signal $\omega_{PLL,I_b}$ remains equal to the electrical pulsation. After several seconds, the frequency estimation reaches zero.

B. CUSUM Algorithm

A theoretical background of the CUSUM algorithm (‘CU-Mulative SUM’) can be found in [18]. This algorithm acts like an integrator that allows detecting variations in signal properties, such as the mean value. The difference of pulsation between the input signal and the electrical pulsation is thus the input of the CUSUM algorithm.
In its simplified version, the algorithm consists in computing a function $g$ at each sample time $k$:

$$ g(k) = \max \left( 0, g(k-1) + \left( \Delta \omega_{x,e}(k) - \frac{\mu_0 + \mu_1}{2} \right) \right) $$

(11)

where $\mu_0$ and $\mu_1$ are the mean value of the signal before and after the fault and $\Delta \omega_{x,e}(k)$ is the input of the CUSUM algorithm (cfr Fig 3). It can be interpreted as follows: before the fault the average value of $\Delta \omega_{x,e}(k) - \frac{\mu_0 + \mu_1}{2}$ will be negative as $\mu_1$ is greater than $\mu_0 = 0$ and $g$ will be maintained to zero. After the fault the term becomes positive and $g$ begins to increase with a slope equal to $\frac{\Delta g}{\Delta x} = \frac{\mu_1 - \mu_0}{2}$. When $g$ will reach a user-chosen threshold $h$, the system will set up a flag error. The value of $\mu_0 = 0$ is straightforward and the theory would impose a dynamic value of $\mu_1 = \omega_e$, what is not convenient. Instead, a constant value can be chosen. A value of $h$ can be roughly calculated as a function of an imposed detection time $\Delta t_{\text{detection}}$ and the minimum speed at which the drive should operate $\omega_{e,\text{min}}$:

$$ h = \frac{\omega_{e,\text{min}} - \frac{\mu_0 + \mu_1}{2}}{t_s} $$

(12)

where $t_s$ is the sampling time.

For, $\mu_1 = 20\text{rad/s}$, $\mu_0 = 0\text{rad/s}$, $\Delta t_{\text{detection}} = 0.2s$, $t_s = 20\mu$s and $\omega_e = 20\text{rad/s}$, it gives $h = 10000$. It has to be noted that this relation is only true for a given speed, but according to (12), the detection time will theoretically decreases when the speed increases. On the other hand, it is difficult to tune the PLL on the whole speed range and for different signal amplitudes, what can cause a long transient before the estimation reaches zero (see Fig. 5). Hence, this transient increases the detection time.

Fig. 6 shows the $abc$ CUSUM functions for the test presented in Fig. 5, i.e. an open-circuit fault in the $a$-phase occurring at time $t = 0.84s$. Test parameters are those mentioned above. The $a$-phase CUSUM function reaches the threshold $h$ at time $t = 0.99s$, i.e. 0.15s after that fault has occurred.

IV. CONTROL STRATEGY

A. Normal operation

A scheme of the voltage control with PI controllers in the $dq0$ reference frame is shown in Fig. 7. The electromagnetic torque reference $T_{em}^*$ is received directly from the user or from a speed controller. The $dq0$ current references are calculated from (8). A compensation of the emf ($E_{dq}$ and $E_0$) is performed. Simple PI controllers are used for driving the $d$- and $q$-axis currents.

The algorithm to calculate the duty cycles $\delta$ is given in [21]:

$$ \delta_x = \frac{V_x^*}{V_{dc}} \; ; \; \delta_n = \frac{V_n^*}{V_{dc}} $$

(13)

where

$$ V_x^* = V_{x}^* + V_{n}^* $$

(14)

and

$$ V_{x0}^* = \text{mid} \left( \frac{-V_{x0}^{n\text{max}}}{2}, \frac{V_{x0}^{n\text{min}}}{2} \right) $$

$$ V_{x0}^* = \text{mid} \left( \frac{-V_{x0}^{n\text{max}}}{2}, \frac{V_{x0}^{n\text{min}}}{2} \right) $$

(15)

B. Control for the post-fault operation

When a fault occurs in one of the three phases, the decision to stop supplying this phase is taken systematically. The machine will then be driven with two phases and the neutral. According to [15] and [16], keeping constant the $d$- and $q$-axis current references is possible if the zero-sequence current is equal to:

$$ I_{0,\text{fault}}^* = \sqrt{2} \left( I_{q0}^* \sin(\theta_e - k \frac{\pi}{3}) - I_{0}^* \cos(\theta_e - k \frac{\pi}{3}) \right) $$

(16)

where $k = \{0, 1, 2\}$ in case of fault in phase $a$, $b$ or $c$ respectively. Open-loop control of the zero-sequence control will then require a zero-sequence voltage reference equal to:

$$ V_{0,\text{fault}}^* = R_s I_{0,\text{fault}}^* + L_0 \frac{d}{dt} I_{0,\text{fault}}^* $$

(17)

V. IMPACT OF FAULTS WITHOUT CONTROL RECONFIGURATION

The impact of the fault can be quantified with different indices such as the torque ripple, the current increase in the remaining phases and the speed/torque deviation from their reference.

A. Single-Phase Open-Circuit Fault

Events leading to a single-phase open-circuit fault are: power electronics fault (after a short-circuit mitigation or gate drive fault) or mechanical fault (broken wire, unscrewed
connector). Response of the classic three-phase drive has been studied in [22]. Torque is pulsating and current increases in the two remaining phases, both amplitudes depending on the current controller. The speed keeps its average value and the speed ripple depends on the inertia of the system.

Fig. 8 shows the $I_{dq0}$ currents for the considered fault and the proposed control and topology. The peak-to-peak torque ripple is reduced with a factor 3 compared to the classic three-phase system. This is due to the neutral connection and because the neutral current is free to flow (open-loop voltage control of the zero-sequence component).

B. Current Sensor Fault

Test results with a $b$-phase current sensor fault at time $t = 0.16s$ is presented in Fig. 9. The real $abc$ phase-currents are shown on the top. The increase of the currents is due to the controllers, which have to impose their reference based on the $a$- and $c$-phase current measurements only. However, the real $b$-phase current is different from zero, what gives an electromagnetic torque higher than expected. The motor speed increases until almost twice the initial speed (middle figure). It has to be noted that the $a$-phase frequency estimation follows the speed measurements very well whereas the $b$-phase frequency estimation reaches zero as expected. The speed controller will then decrease the torque reference to restore the steady-state. This explains why the $q$-axis current seen from the controller slowly decreases (bottom figure).

VI. FDI AND CONTROL RECONFIGURATION

A. Single-Phase Open-Circuit Fault

Fig. 10 shows the test results where a single-phase open-circuit fault occurs at time $t = 0.23s$ in the $a$-phase. The $a$ CUSUM function reaches the threshold at time $t = 0.83s$. Operating conditions are: $\Omega_m = 300rpm, T_{em} \approx 5Nm$. We can observe that the control reconfiguration reduces the $q$-axis current ripple to nearly zero.

B. Current Sensor Fault

Fig. 11 shows the test results where a current sensor fault occurs at time $t = 0.31s$ in the $b$-phase. The $b$ CUSUM function reaches the threshold at time $t = 0.42s$. Operating conditions are: $\Omega_m = 150rpm, T_{em} \approx 2.8Nm$. We can observe that the control reconfiguration reduces again the $q$-axis current ripple to nearly zero. A comparison with the single-phase open-circuit fault shows that: in case there is no open-circuit fault, the current is still able to flow in the machine and is then not exactly equal to zero (top figure). The speed still increase when the fault occurs. However, the electromagnetic torque is restored with the control reconfiguration and the speed controller must not compensate the error anymore. Another main difference is that the detection time is much lower for the second test. This is due to the PLL transient response that takes more time for higher speeds.
VII. CONCLUSION

A PLL-based software fault detection and isolation for single-phase open-circuit and current sensor faults has been presented and validated with experimental results. The fault diagnosis is used to reconfigure the control by taking advantage of the fault-tolerant topology. A systematic decision to stop supplying the faulty phase has been adopted.

ACKNOWLEDGMENT

Fabien Meinguet would like to thank the Belgian Fund for training in Research in Industry and in Agriculture (F.R.I.A.) for the financing of this research project.

REFERENCES


