Hardware Implementation of FTC of Induction Machine on FPGA

S. Boukadida, S. Gdaim, and A. Mtibaa

***Abstract*—In this paper, a new design method of Direct Torque Control using Space Vector Modulation (DTC-SVM) of an Induction Machine (IM), which is based on Fault Tolerant Control (FTC) is proposed. Due to its complexity, the FTC implemented on a microcontroller and a Digital Signal Processor (DSP) is characterized by a calculating delay. To solve this problem, an alternative digital solution is used, based on the Field Programmable Gate Array (FPGA), which is characterized by a fast processing speed. However, as an FPGAs increase in size, there is a need for improved productivity, and this includes new design flows and tools. Xilinx System Generator (XSG) is a high-level block-based design tool that offers bit and cycle accurate simulation. This tool can automatically generate the Very High-Density Logic (VHDL) code without resorting to a tough programming, without being obliged to do approximations and more we can visualize the behavior of the machine before implementation which is very important for not damage our machine. Simulation and experimental results using Hardware In the Loop (HIL) of the FTC based DTC-SVM is compared with those of the conventional DTC. The comparison results illustrate the reduction in the torque and stator flux ripples. Our purpose is to reveal our algorithm efficiency and to show the Xilinx Virtex V FPGA performances in terms of execution time.**

***Index Terms*—FPGA, Fault tolerant control, induction motor, parity space, Xilinx System Generator.**

Original Research Paper
DOI: 10.7251/ELS1620076B

I. Introduction

Nowadays, the implementation of modern controllers for electrical drives demands a perfect satisfaction of the required performances. In order to satisfy control requirements, recent research studies proved that digital hardware solutions, such as FPGA, are an appropriate alternative to software solutions. Research interest in implementation on FPGA has grown considerably over the past few years. This is due to its advantages such as reducing the execution time by adopting a parallel processing and the rapid prototyping of digital control.

The cost of FPGA decreases, for that reason, the replacement of microcontrollers with FPGA is a new tendency. During the past few years, several researchers use the FPGA for controlling electrical system [1-5]. Most of them develop the algorithm on a VHDL hardware description language. If the prototyping platform uses an FPGA to run the algorithm, a newly created simulation tool can be used not only to simulate exactly the hardware but also to automatically generate the VHDL code needed for the implementation. This software is the tool XSG developed by Xilinx and integrated in MATLAB/SIMULINK.

FTC of electrical drives was a very active research field for many research groups [6-9]. FTC should be able to detect faults and to cancel their effects or to attenuate them until an acceptable level. The FTC aims to ensure the continuous system functionality, even after faults occurrence. This allows increasing system availability and reliability. Different types of failures can occur in controlled electrical drives: IM, power converters, connectors, and sensors. The failures in the electric motor can have various origins such as failures related to the exploitation that can lead to faults and also a premature degradation, and failures related to wrong weak dimensioning and design which lead to a premature degradation [10]. Current sensors are widely used in controls of electrical drives. Faults in the current measurement chains have been treated for various electrical drives. Researches in this field initially focused on the effect of current sensor fault and the development of Fault Detection and Isolation (FDI) methods [11-14]. In general, FDI methods utilize the concept of redundancy, which can be either a hardware redundancy or analytical redundancy. The usual approach to fault diagnosis is based on hardware redundancy and uses a voting technique to decide if a fault has occurred and to locate it among the redundant system elements. Also, the analytical redundancy FDI approach makes use of a mathematical model of the monitored system. The analytical redundancy approach does not require additional hardware, it is usually a more cost-effective approach compared to the hardware redundancy approach. However, the analytical redundancy approach is more challenging due to the need to ensure its robustness in the presence of model uncertainties, noise, and unknown disturbances. Generally, the analytical redundancy approach can be divided into quantitative model-based methods and

Manuscript received on March 12, 2016. Received in revised form on November 2, 2016. Accepted for publication on November 10, 2016.

B. Souha is with Laboratory of Electronics and Microelectronics, Faculty of Sciences, University of Monastir, Tunisia, e-mail: boukadidasouha@gmail.com.

S. Gdaim is with Laboratory of Electronics and Microelectronics, Faculty of Sciences, University of Monastir, Tunisia, e-mail: sgdaim@yahoo.fr.

A. Mtibaa is with Laboratory of Electronics and Microelectronics, Faculty of Sciences, University of Monastir, Tunisia, e-mail: abdellatif.mtibaa@enim.rnu.tn.

qualitative model-based methods. The quantitative model-based methods, such as the observer-based methods [15], use explicit mathematical models and control theories to generate residuals for FDI. On the other hand, the qualitative model-based methods use artificial intelligence techniques [16]. The detection and location of defects are essential but not sufficient to ensure the safety and operation in degraded mode. Researchers now are more concerned with FTC strategies based on reconfigurable controls in order to guarantee continuous operation of the system, even with a sensor failure [17-20].

Digital implementations of FTC are mostly carried out with microcontrollers or digital signal processors owing to their software flexibility and low cost. FPGA can also be used as a new digital solution. In this paper, authors propose a new and simple FTC current that allows continuous operation of a three-phase controlled machine drive under fault current measurement. The adopted control method uses three current sensors instead of only two as usually done. Residuals for FDI are then provided via analytical redundancy introduced by the third sensor. In the case of current sensor fault occurrence, the faulty current sensor is eliminated and the control continues working with the two other healthy sensors. For the hardware implementation of the FTC of an IM on the FPGA, we use XSG toolbox developed by Xilinx and added to Matlab/Simulink. The XSG advantages are the rapid time to market, real time and portability. Once the design and simulation of the proposed algorithm are completed we can automatically generate the VHDL code in Xilinx ISE.

This paper is organized as follows. In Section II, the new design methodology for implementation on FPGA is presented. In section III, the design of the DTC of IM using SVM is introduced. In Section IV, the FTC approach is developed. The implementation of the proposed architecture is presented in the next section. Finally, a conclusion closes this paper.

II. Design Ethodology for Implementation on FPGA

The design developed in this paper was performed according to an appropriate methodology. The XSG is a modeling tool developed by Xilinx. This tool can automatically generate the VHDL code without resorting to a difficult and tough programming. XSG facilitates the implementation of any design and it eases the designer from low-level algorithmic complexity. XSG uses the Xilinx Integrated Synthesis Environment design suite to automate VHDL code generation which can then be integrated with other designs or used as a stand-alone design. Unlike the VHDL languages, the new methodology (XSG) gives a model-based design interface using an extended Simulink library of building blocks to create hardware. The XSG library has a set of DSP hardware blocks that can perform complex functions such as Interpolation filter, Cordic (divider, tan, sin, cos, Log), FFT and FIR filter design. This new methodology consists of a set of steps and roles that offer considerable hardware design advantages as shown in Fig. 1.

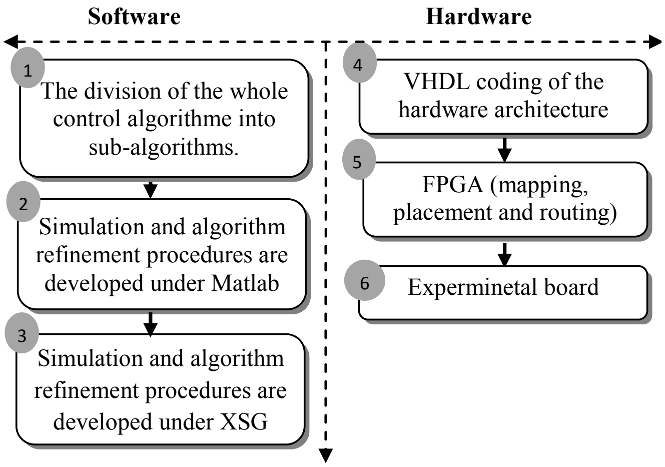

Software
Hardware
1
The division of the whole control algorithme into sub-algorithms.
2
Simulation and algorithm refinement procedures are developed under Matlab
3
Simulation and algorithm refinement procedures are developed under XSG
4
VHDL coding of the hardware architecture
5
FPGA (mapping, placement and routing)
6
Experminetal board

Fig. 1. Design methodology for implementation on FPGA

System Generator gives hardware co-simulation, making it possible to incorporate a design running on an FPGA directly into a Simulink simulation. Hardware co-simulation compilation targets automatically create a bit stream and associate it with a block. HIL, or FPGA in the loop, is a concept that as revealed by the name uses the hardware in the simulation loop. This leads to easy testing and the possibility to see how the actual plant is behaving in hardware. By having the stimuli in software on the PC, implementing a part of the loop in hardware and then receiving the response from hardware back in the software.

III. Design of the DTC of IM Using SVM

A. Conventional Direct Torque Control of IM

Figure 2 presents a possible schematic of DTC. IM is the system to be controlled.

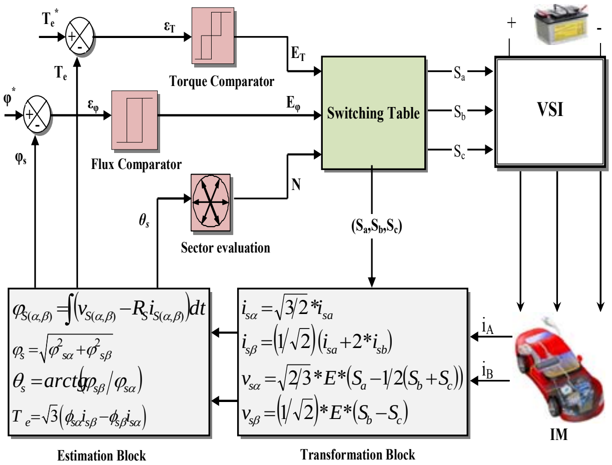

T_e^*
T_e
ε_T
Torque Comparator
E_T
φ^*
φ_s
ε_φ
Flux Comparator
E_φ
Switching Table
S_a
S_b
S_c
VSI
+
-
θ_s
Sector evaluation
N
(S_a,S_b,S_c)
$\varphi_{S(\alpha,\beta)}=\int(v_{S(\alpha,\beta)}-R_S i_{S(\alpha,\beta)})dt$
$\varphi_s=\sqrt{\varphi_{s\alpha}^2+\varphi_{s\beta}^2}$
$\theta_s=arctg(\varphi_{s\beta}/\varphi_{s\alpha})$
$T_e=\sqrt{3}(\phi_{s\alpha}i_{s\beta}-\phi_{s\beta}i_{s\alpha})$
Estimation Block
$i_{s\alpha}=\sqrt{3/2}*i_{sa}$
$i_{s\beta}=(1/\sqrt{2})(i_{sa}+2*i_{sb})$
$v_{s\alpha}=\sqrt{2/3}*E*(S_a-1/2(S_b+S_c))$
$v_{s\beta}=(1/\sqrt{2})*E*(S_b-S_c)$
Transformation Block
i_A
i_B
IM

Fig. 2 . Diagram of the DTC method.

Schematic of DTC is composed of two different loops corresponding to electromagnetic torque and stator flux. Two hysteresis comparator of torque and flux are used. The outputs of the stator flux error and torque error hysteresis blocks, together with the position of the stator flux are used as inputs

to the switching table given in TABLE I.
The module of the stator flux φ_s is given by equation 1, the developed electromagnetic torque T_e is evaluated by equation 2 and the shifted angle θ_s is presented by equation 3.

TABLE I
THE SWITCHING TABLE FOR BASIC DTC

		Number of sectors					
E_φ	E_T	S1	S2	S3	S4	S5	S6
1	1	V2	V3	V4	V5	V6	V1
	0	V7	V0	V7	V0	V7	V0
	-1	V6	V1	V2	V3	V4	V5
0	1	V3	V4	V5	V6	V1	V2
	0	V0	V7	V0	V7	V0	V7
	-1	V5	V6	V1	V2	V3	V4

$$\varphi_s = \sqrt{\varphi_{s\alpha}^2 + \varphi_{s\beta}^2} \tag{1}$$

$$T_e = \sqrt{3}\left(\phi_{s\alpha} i_{s\beta} - \phi_{s\beta} i_{s\alpha}\right) \tag{2}$$

$$\theta_s = \tan^{-1}\left(\frac{\phi_{s\beta}}{\phi_{s\alpha}}\right) \tag{3}$$

Where φ_s and i_s denote stator flux and stator currents, in reference (α, β), respectively.

B. DTC Inconvenience

The DTC algorithm is the most recently developed technique of IM. This drive was firstly proposed by Takahashi. The DTC method is characterized by a fast dynamic response, robustness to the rotor parameter variation and by its simple implementation. However, this technique suffers from many drawbacks. Among these disadvantages are stator flux and electromagnetic torque ripples. Thus, ripples can be observed on the controlled torque and flux which can be reflected on the driving shaft and caused damage to the structure. It is well established that these drawbacks are mainly due to the use of hysteresis torque and flux controllers. For this reason, many methods are used to overcome these disadvantages such as SVM. To overcome these disadvantages of the conventional DTC, in the next section, we present and discuss the hardware implementation of the DTC using SVM on the FPGA.

C. Direct Torque Control of IM-based SVM

In this section, a DTC-SVM approach is proposed to reduce torque and flux ripples. Fig.3 illustrates the proposed version of the DTC-SVM with the IM. In order to have a switching frequency operation fixed, we combine the DTC with SVM. The switching table and the two hysteresis controller are replaced by two PI regulators, transformation block (d, q) to (α, β) and a SVM block. The outputs of the regulator are used as inputs of direct axis voltage. The (d, q) axis voltages are converted into amplitude of stator voltage using equation 4.

$$\begin{aligned} V_{s\alpha} &= V_d \cos(\theta_s) - V_q \sin(\theta_s) \\ V_{s\beta} &= V_d \sin(\theta_s) + V_q \cos(\theta_s) \end{aligned} \tag{4}$$

The voltages (V_d, V_q) and stator flux angle Θ_s are used as reference signals in SVM approach.

The principle of SVM is to project the desired stator voltage vector on the two adjacent vectors corresponding to two switching states of the inverter. The values of these projections provide the desired commutation times. The coordinates of the voltage vector in the base formed by adjacent vectors are calculated using equation 5. To apply the reference vector, it is then possible to apply the vector V_1 during a time t_1 and the vector V_2 during a time t_2.

$$\overrightarrow{V_s} = V_{s\alpha} + jV_{s\beta} = \frac{T_1}{T_{\text{mod}}}\overrightarrow{V_1} + \frac{T_2}{T_{\text{mod}}}\overrightarrow{V_2} \tag{5}$$

Where T_{mod} is the switching period, T_1 is the time of application of V_1 and T_2 is the time of application of V_2.

The key step of the SVM technique is the determination of T_i ($1 \leq i \leq 3$) during every modulation period T_{mod}. Expanding equation 10, it is possible to express the time T_1 and T_2 in terms of $V_{s\alpha}$ and $V_{s\beta}$. The conduction time will be expressed as follows:

$$\begin{aligned} T_1 &= \left(\sqrt{\frac{3}{2}}.V_{s\alpha} - \sqrt{\frac{1}{2}}V_{s\beta}\right).\frac{T_{\text{mod}}}{E} \\ T_2 &= \sqrt{2}.V_{s\beta}.\frac{T_{\text{mod}}}{E} \end{aligned} \tag{6}$$

Where E denote continuous voltage of the inverter.

To facilitate the calculations, we normalize the voltages $V_{s\alpha}$ and $V_{s\beta}$ by posing:

$$\begin{aligned} D_1 &= \frac{\sqrt{3}}{\sqrt{2}}.\frac{V_{s\alpha}}{E} - \frac{1}{\sqrt{2}}\frac{V_{s\beta}}{E} \\ D_2 &= \sqrt{2}\frac{V_{s\beta}}{E} \\ D_0 &= 1 - D_1 - D_2 \end{aligned} \tag{7}$$

The duties of each phase of the inverter are presented as follows:

$$\begin{aligned} S_a &= 0.5(1 + D_1 + D_2) \\ S_b &= 0.5(1 - D_1 + D_2) \\ S_c &= 0.5(1 - D_1 - D_2) \end{aligned} \tag{8}$$

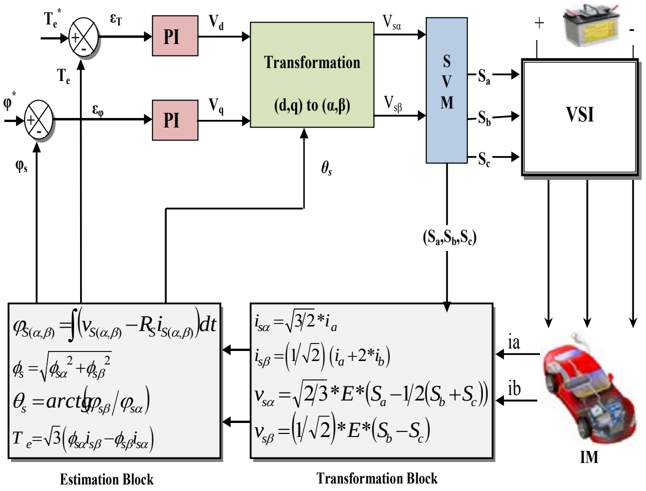

T_e^*
ε_T
PI
V_d
Transformation
(d,q) to (α,β)
$V_{s\alpha}$
S V M
+
-
T_e
φ^*
ε_φ
PI
V_q
$V_{s\beta}$
S_a
S_b
S_c
VSI
φ_s
θ_s
(S_a, S_b, S_c)
$\varphi_{S(\alpha,\beta)} = \int (v_{S(\alpha,\beta)} - R_S i_{S(\alpha,\beta)}) dt$
$\phi_s = \sqrt{\phi_{s\alpha}^2 + \phi_{s\beta}^2}$
$\theta_s = arctg(\varphi_{s\beta}/\varphi_{s\alpha})$
$T_e = \sqrt{3}(\phi_{s\alpha} i_{s\beta} - \phi_{s\beta} i_{s\alpha})$
$i_{s\alpha} = \sqrt{3/2} * i_a$
$i_{s\beta} = (1/\sqrt{2})(i_a + 2 * i_b)$
$v_{s\alpha} = \sqrt{2/3} * E * (S_a - 1/2(S_b + S_c))$
$v_{s\beta} = (1/\sqrt{2}) * E * (S_b - S_c)$
ia
ib
IM
Estimation Block
Transformation Block

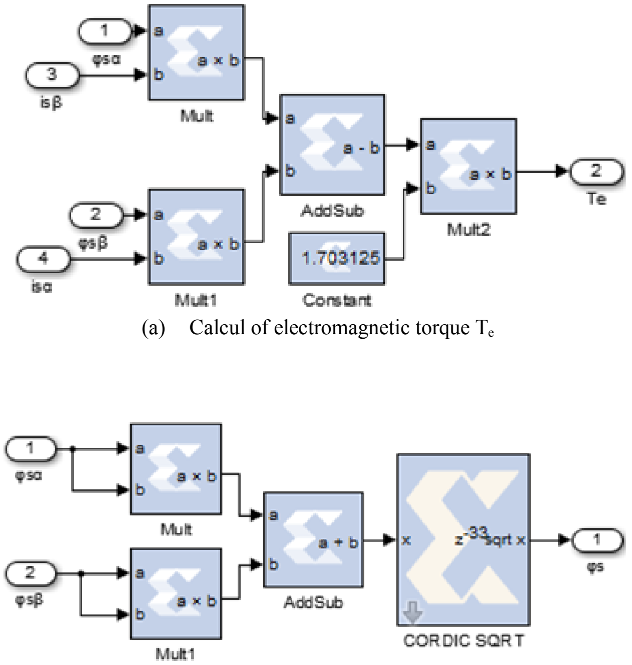
Fig. 3. Diagram of the DTC-SVM method.

D. Modeling of proposed DTC-SVM scheme using XSG

XSG tool developed for MATLAB/SIMULINK package is widely used for verification purposes and algorithm development in FPGA and DSP. The main advantage of this tool is to translate the modeling design of DTC-SVM scheme into hardware implementation and to generate the VHDL code without restoring to a difficult programming. Furthermore, the FPGA hardware represents the blocks of the XSG tool developed for MATLAB/SIMULINK package. Therefore, the implementation time is reduced because the algorithm needs to be simulated and modeled just once. The design of the DTC-SVM using the XSG is based on a mathematical model. The design stages for logical operations and required arithmetic for the proposed design are carried out in a modular fashion and hierarchical. The modules of proposed design can be described as follows:

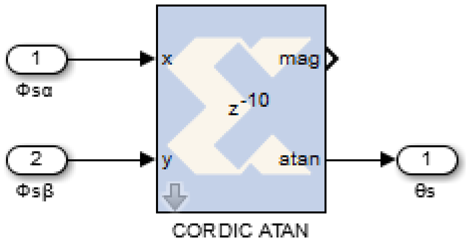
- Torque, flux Estimator and angle calculation :

The XSG design of torque and flux estimator module and its submodules of proposed DTC-SVM are shown in Fig. 4. The module of the stator flux φ_s given in equation 1, the developed electromagnetic torque T_e given in equation 2 and the shifted angle θ_s presented by equation 3 are modeled using blocks of XSG. The XSG library contains many blocks that are used to develop our design such as; basic elements (multiplexers, delay, registers); mathematical functions (multiplication, relational, add, negate).



(a) Calcul of electromagnetic torque T_e

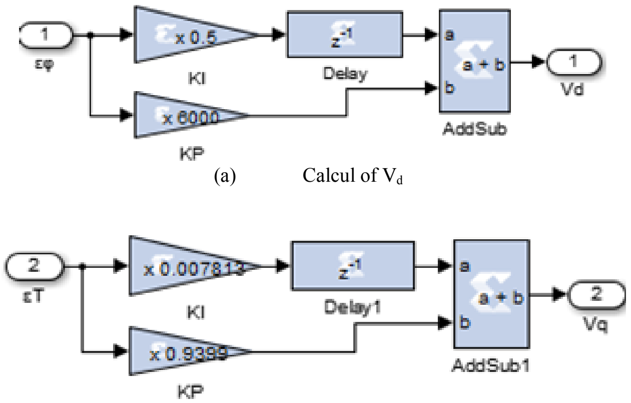
(b) Calcul of stator flux φ_s



(c) Calcul of angle

Fig. 4. Modeling of torque, flux and teta calculating

The XSG design of the (d, q) axis voltages are shown in Fig.5.



(a) Calcul of V_d

(b) Calcul of V_q

Fig. 5. Calcul of (V_d, V_q)

- Transformation block (d, q) to (α, β)

The outputs of the regulator are used as inputs of direct axis voltage, as shown in Fig.6.

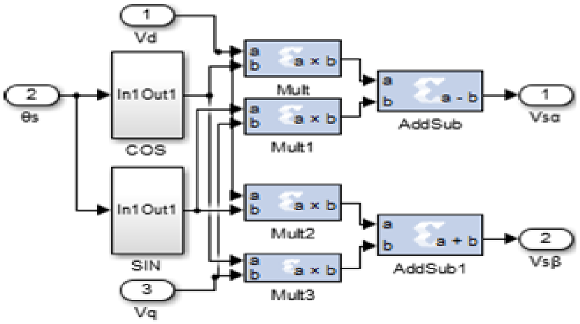


Fig. 6 Design of the components $V_{s\alpha}$ and $V_{s\beta}$

- Space Vector Modulation.

The block SVM generates a series of pulses to be used subsequently to carry out the control signals used in the model of the inverter as shown in Fig.7.

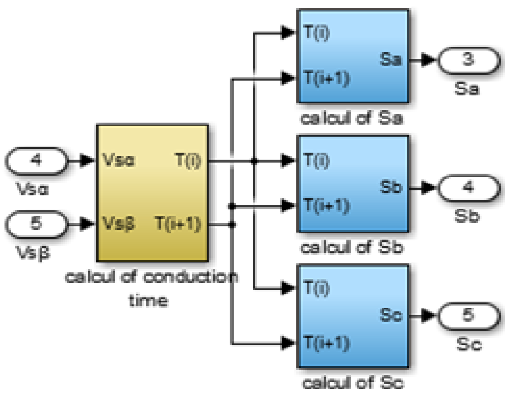


Fig. 7. Calcul of S_a, S_b and S_c

IV. OVERVIEW OF FDI AND RECONFIGURATION TECHNIQUES

In this section, the dynamic redundancy equations based on the Parity Space (PS) approach with the proposed FDI algorithm are first presented. Then the control reconfiguration is presented.

A. Proposed FDI Algorithm

The concept of PS utilizes the information ingrown in the mathematical model for FDI. The actual behavior is compared to that expected on the basis of the model; deviations are indications of faults (or disturbances, noise or modeling errors). Primary residuals are formed as the difference between those predicted by the model and the actual plant outputs. These are then subjected to a linear transformation, to obtain the desired FDI properties.

The aim of this study is to ensure a quasi-instantaneous detection of fault occurrence on the sensor in order to ensure continuous operation.

A brief description of the PS approach is given here:

$$\frac{dx}{dt} = A.X(t) + B.u(t) \quad ; \; A = (a_{jq})_{\,1\le j,q\le n} \tag{9}$$

$$\begin{gathered} y(t) = C.X(t) + f(t); B = (b_{jr}) \\ 1 \le j \le n \; and \, 1 \le r \le m \end{gathered} \tag{10}$$

Where X(t) denotes the state vector, u(t) the control input vector and y(t) the measured output vector. All sensor faults are grouped in the f (t) term, which is generally unknown. Matrices A, B and C are the known matrices that depend on coefficients a_{jq} and b_{jr}.

In this study, the measurement equation of each sensor is presented as:

$$y_i(t) = C_i.X(t) + f_i(t); C_i = (0.....1........0) \tag{11}$$

The discrete form of (9) and (10) is expressed by (12), where T_a is the sampling time of measurement:

$$\begin{gathered} X(k+1) = A_d.X(k) + B_d.u(k) \\ y_i(k) = C_{di}.X(k) + f_i(k) \end{gathered} \tag{12}$$

Where:

$$A_d = e^{A.T_a} \; ; B_d = B.T_a \; ; C_d = C_i \tag{13}$$

Using temporal redundancy, the system (12) and (13) is expressed as:

$$\begin{pmatrix} y_i(k) \\ y_i(k+1) \\ . \\ . \\ y_i(k+s) \end{pmatrix} - \begin{pmatrix} 0 & 0 & 0 & 0 \\ C_iB & 0 & ... & .. \\ C_iAB & C_iB & ... & .. \\ .... & ... & & \\ C_iA^{s-1}B & C_iA^{s-2}B & ... & 0 \end{pmatrix} . \begin{pmatrix} u(k) \\ u(k+1) \\ . \\ . \\ u(k+s) \end{pmatrix} = \begin{pmatrix} C_i \\ C_iA \\ . \\ . \\ C_iA^s \end{pmatrix} X(k) + F_i(k,s) \tag{14}$$

The residual generation using PS is defined by:

$$\begin{aligned} r_i(k) &= V_i\,(Y_i(k,s) - G_i(s)U(k,s) \\ &= V_i\,(H_i(s)X(k) + F_i(k,s)) \\ &= V_i\,F_i(k,s) \end{aligned} \tag{15}$$

Where: H_i(s) is the observability matrix of “s” order, Y_i and U are constructed using temporal redundancies of the known input and output, G_i (s) is the control matrix, Matrix F_i is related to faults, V_i is a projection vector, which is derived from the following relation:

$$V_i.H_i = 0 \tag{16}$$

The V_i vector parity is (1×3) dimension and is defined by (17). Many solutions are available for $V_i.H_i = 0$. This leads to freedom degrees V_i parameters choice. One solution is given by (18).

$$V_i = \left[V_1 \; V_2 \; V_3\right] \tag{17}$$

$$V_i = \left[A^2 \; -2A \;\; 1\right] \; when \; A = 1 - \frac{1}{\alpha}T_a \tag{18}$$

Where α is the current time constant.

In electrical drives, sampling time T_a can be very small (3 to 5μs are typical values for high-performance systems), while current time constant is generally higher than 100μs. The approximation is applied; the derived simple vector projection V_i and the residual r are expressed by:

$$V_i = \left[1 - 2\ \ 1\right] \tag{19}$$

$$r = y_0 - 2.y_1 + y_2 \tag{20}$$

To make residual variations when a fault occurs more significant and easier to detect, the absolute value is considered:

$$r = \left| y_0 - 2.y_1 + y_2 \right| \tag{21}$$

Residual depends only on sensor outputs; therefore, it can be applied to any system with fast reading acquisition, independently of the complexity of the system model.

In this paper, without the necessity of a decision algorithm as most of FDI methods developed in the literature, the residual allows the isolation of a faulty sensor directly and it does not depend on any system parameters.

B. Proposed FTC Algorithm

In order to obtain high-performance motor drives, a modern control strategy like DTC-SVM is employed. This technique is inherently dependent on the measurement sensors, which should operate properly. However, when these sensors fail, the control system needs to compensate for the failure to function properly. This necessitates the backup systems to support the proper operation of the drive in case of sensor failure.

The adopted control method uses three current sensors instead of only two as usually done. Residuals for FDI are then provided via analytical redundancy introduced by the third sensor. In the case of current sensor fault occurrence, the faulty current sensor is eliminated and the control continues working with the two other healthy sensors.

Assuming that the controlled induction motor drive is with an isolated neutral ($I_a + I_b + I_c = 0$), the reconfigurable currents (I_{ar}, I_{br} and I_{cr}) can be computed in different ways as presented in TABLE II.

TABLE II
FAULT INDICATOR ACCORDING TO THE FAULTY SENSOR

Faulty Current	I_{ar}	I_{br}	I_{cr}
No fault	I_a	I_b	I_c
I_a	$-(I_b+I_c)$	I_b	I_c
I_b	I_a	$-(I_a+I_c)$	I_c
I_c	I_a	I_b	$-(I_a+I_b)$

V. XSG SIMULATION RESULTS OF PROPOSED SCHEME

The main objective of this paper is to design an easy and simple FTC that is based on PS and implements it on a Xilinx Virtex 5 XC5VFX50T-1FFG1136 (ML506) FPGA. The first step consists of building of the control algorithm with the use of XSG blocks. The modeling system is then connected to the simulated power system by the Gateway "In" and Gateway "Out" blocks, as shown in Fig.8. Once the design of the system is completed and gives the desired simulation results, the VHDL code can be generated by the XSG tool. After a generation of VHDL code and the synthesis, we can generate the bit stream file. Then we can move this configuration file to program the FPGA.

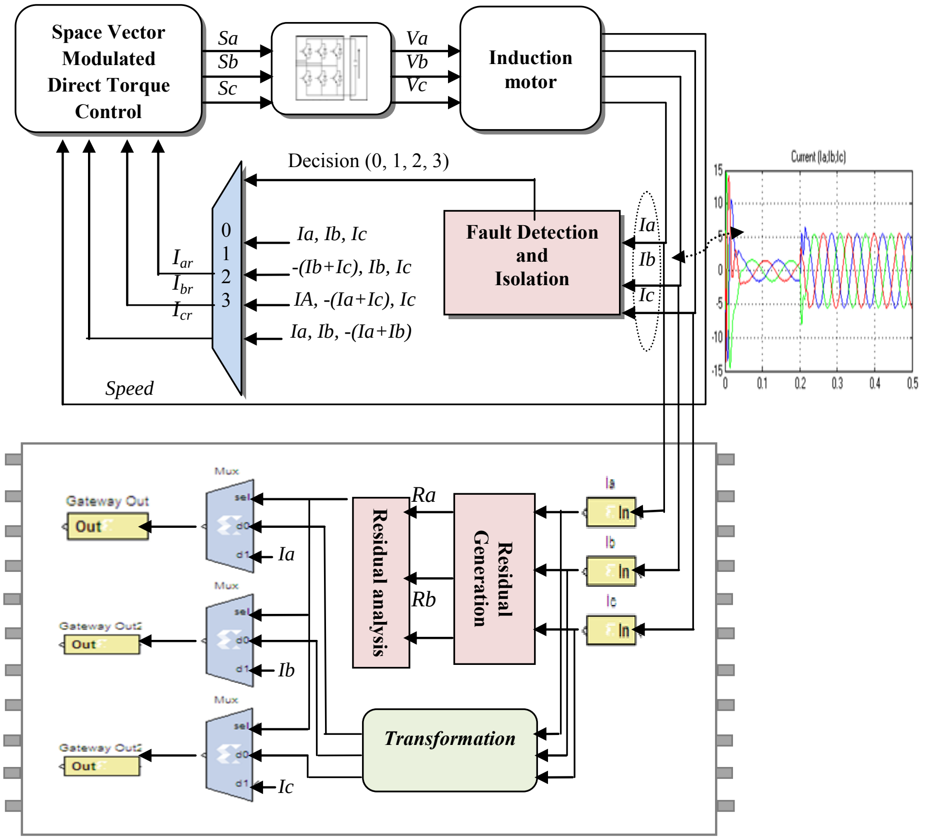


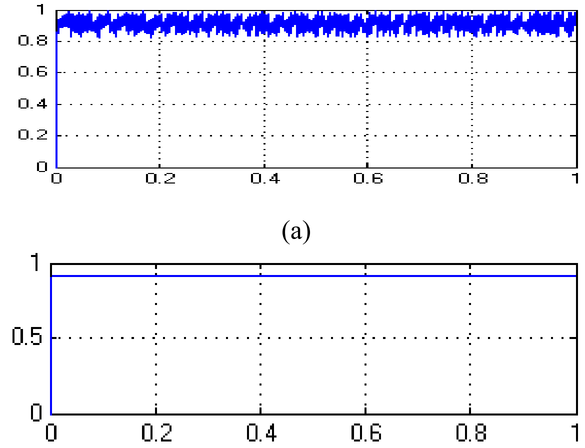
Fig. 8. Proposed Fault-tolerant control

Control system's algorithm must be functionally validated before implementation. Results using hardware co-simulation is presented to assess the ability of this diagnostic approach to detect isolate and reconfigure sensor faults in an IM. In order to respect technical constraints of the power inverter, the sampling period is 50 µs. The machine is running at 300 rad/sec. The flux and torque references used are 0.91 Wb and 10 N.m respectively.

In Fig. 9 (a and b) is illustrated that compared with the conventional DTC, ripples of stator flux is reduced.

The simulation results in Fig.10 (a and b) shows that the torque's ripples with DTC-SVM in steady state are reduced significantly compared with conventional DTC.

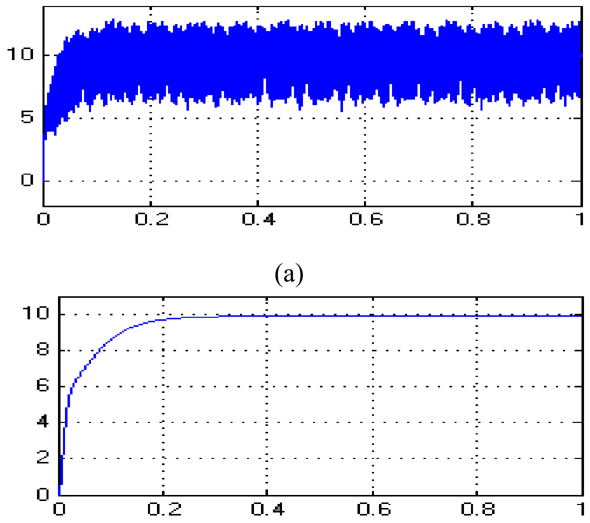
In all simulation presented can be observed a significant better behavior of the performance achieving the main objectives of the present work which was to reduce the torque ripple and maintain a good torque response as the conventional DTC. The performance of the control system is improved.



(a)

(b)

Fig. 9. Evolution flux for: (a) Conventional DTC (b) DTC-SVM



(a)

(b)

Fig. 10. Evolution torque for: (a) Conventional DTC (b) DTC-SVM

Fig. 11 shows the residual evolution when a current sensor fault occurs (at t = 1.5 s) in I_a. The measured current changes in a discontinuously, and this change is effectively detected by the FDI algorithm as a sensor fault. In order to get clear curves, magnitude of faults is chosen relatively high: a 2.5 offset error. Faults are applied at t=1.5s, where $I_{amax} = 5A$. The maximum value of residual is $R + 2d \approx 0.1+2\times 2.5 \approx 5$ for an offset fault. When a fault occurs on one AC current sensor, I_a, I_b or I_c, the corresponding residual R_a, R_b or, R_c, respectively, becomes superior to the defined threshold, ε. In this case, the threshold was set equal to 0.1A.

Fig. 12 shows the measured current, with faulty sensor, reconfigured current, and the fault indicator D_i. After a fault occurrence, the FTC is very rapidly reacting to prevent any undesirable event such as cascaded failures. Therefore, the time duration of the fault detection and the execution time of the backup strategy are very small as illustrated in Fig. 12. The quality and the quickness of the transition between the healthy operation mode and the faulty one are demonstrated.

VI. SYNTHESIS RESULTS OF PROPOSED SCHEME

The main objective of this paper is to design an easy and simple FTC that is based on PS and implement it on a Xilinx Virtex 5 XC5VFX50T-1FFG1136 FPGA. The whole design is developed under XSG environment. In the first step, we begin by implementing the proposed architectures using the XSG blocks available on the Simulink library. Once the design of the system is completed and gives the desired simulation results, the VHDL code can be generated by the XSG tool. After a generation of VHDL code and the synthesis, we can generate the bit stream file. Then we can move this configuration file to program the FPGA.

System Generator gives hardware co-simulation, making it possible to incorporate a design running on an FPGA directly into a Simulink simulation. Hardware co-simulation compilation targets automatically create a bit stream and associate it with a block. Hardware in the loop (HIL), or FPGA in the loop, is a concept that as revealed by the name uses the hardware in the simulation loop. This leads to easy testing and the possibility to see how the actual plant is behaving in hardware. By having the stimuli in software on the PC, implementing a part of the loop in hardware and then receiving the response from hardware back in the software.

The synthesis is the process of transforming one representation in the design abstraction hierarchy to another representation. In this step, the VHDL codes are synthesized to be converted into gate level view of the FTC architecture. Resource utilization of FTC implementation on FPGA is shown in Table III. It presents the information concerning the number of Input Output blocks, Slices Registers, Slice LUTs and number of DSP. The performance of the hardware solution based on the FPGA in terms of execution time is shown in Table IV.

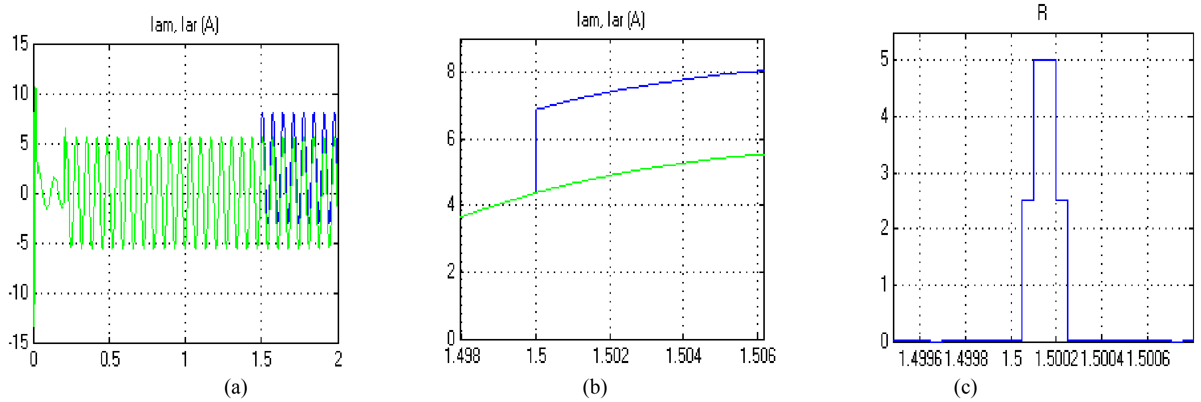


Fig. 11. (a): Measured and reconfigured current ; (b, c) zooms at fault application time.

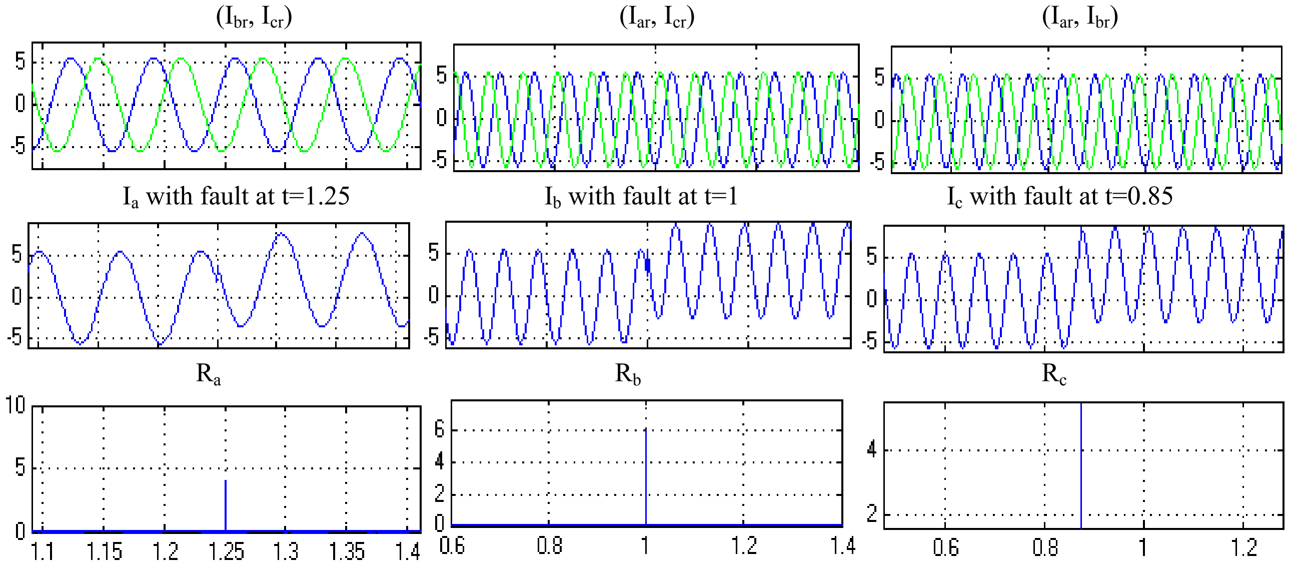


Fig. 12. Results of the reconfigured i(a ,b , c)$_r$, measured currents i(a ,b , c) and residual.

TABLE III
USED RESOURCES

	Used	Available
Number of bonded IOBs	203	640
Number of Slices Registers	828	44800
Number of Slice LUTS	27364	44800
Number of DSP48Es	124	128

TABLE IV
FPGA TIME PERFORMANCE OF THE FTC

MODULE	EXECUTION TIME (μS)
DTC-SVM	0.88
FDI	0.15
FTC	0.42
AD INTERFACE	2.4

The performances of the hardware solution based on the FPGA compared to software solutions: Digital Signal Processor dSPACE and microcontroller, is shown in Fig. 13.

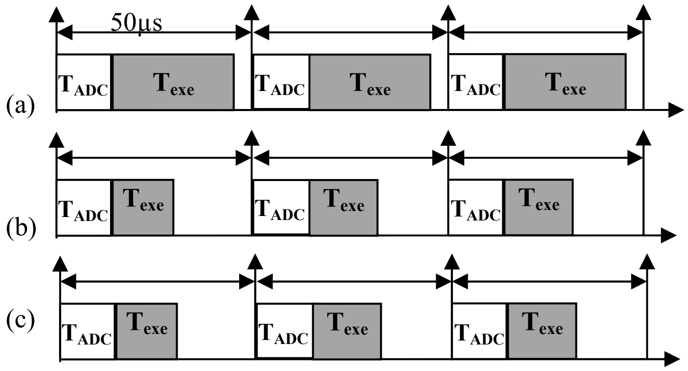


Fig. 13. Timing diagram for the implementation on: (a) microcontroller STM (b) dSPACE (c) Xilinx Virtex–V FPGA

Where T_{ADC} is Analogue to digital conversion time and T_{exe} is Temps execution.

The whole exection time control of the control algorithm is equal to:

$$T_{exe}= T_{DTC\text{-}SVM} + t_{FDI} + t_{SCSR}$$
$$= 0.88 + 0.15 + 0.42= 1.45$$

In [21], this paper deals with experimental validation of a reconfiguration strategy for sensor fault-tolerant control in IM. The proposed active FTC is implemented using a dSPACE 1103. In paper [22], the execution time is of 300µs using the dSPACE 1102. Using the dSPACE, the sampling time is to 100µs, due to the sequential processing of the dSPACE [23-26]. In this paper, using the FPGA the execution time of the control algorithm of IM is (1 to 2µs). Therefore, the obtained execution time using the FPGA is far lower compared to the software solutions.

VII. Conclusion

In this paper, we propose a new FTC algorithm, through system reconfigurations, under certain faulty sensor conditions. The purpose is to reduce the safety hazards risk, in case of faulty sensors. The design is derived from a PS approach and is based on temporal redundancies. The developed FTC is available for sensors measuring variables, with gradual change. Sudden faults are detected, even low magnitude faults. Simulation results show the performances of the proposed control strategy in terms of ripples of the electromagnetic torque and the stator flux. The implementation results show the performances of the hardware implementation in terms of design time which is reduced, low execution time, and minimal resources utilization.

References

[1] B.A.Welchko, T.A .Lipo., T.M Jahns,., S.E.Schulz, ‘Fault tolerant three-phase AC motor drive topologies a comparison of features, cost, and limitations’, IEEE Trans. Power Electron., 2004, 19, (4), pp. 1108–1116.

[2] H. Jafari, J. Poshtan," Fault Isolation and Diagnosis of Induction Motor Based on Multi-Sensor Data Fusion", International Power Electronics Drive Systems and Technologies Conference (PEDSTC20 15) 3-4 February 2015, Tehran, Iran.

[3] H. Berriri, M. Wissem Naouar, I. Slama-Belkhodja, “Easy and Fast Sensor Fault Detection and Isolation Algorithm for Electrical Drives", IEEE transactions on power electronics, vol. 27, no. 2, February 2012.

[4] A. Adouni, M. Ben hamed, A. Flah, L. Sbita," Sensor and actuator Fault Detection and Isolation based on Artificial Neural Networks and Fuzzy logic applicated on induction motor", International Conference on Control, Decision and Information technologies 2013 pp917-922.

[5] T. Abbasian Najafabadi, F.R.Salmasi, P. Jabehdar-Maralani," Detection and Isolation of Speed-, DC-Link Voltage-, and Current-Sensor Faults Based on an Adaptive Observer in Induction-Motor Drives", IEEE transactions on industrial electronics, vol. 58, no. 5, may 2011.

[6] H.Alwi, C. Edwards, C.Pin Tan, "Fault Detection and Fault-Tolerant Control Using Sliding Modes”, Advances in Industrial Control, 2011.

[7] K. M. Passino, P. J. Antsaklis, "Fault detection and identification in an intelligent restructurable controller", Journal of Intelligent and Robotic Systems 1988, Volume 1, Issue 2, pp 145-161.

[8] M. Wissem Naouar, E. Monmasson, A. A. Naassani and I. S.-Belkhodja, "FPGA-Based Dynamic Reconfiguration of Sliding Mode Current Controllers for Synchronous Machines”, IEEE transactions on industrial informatics, vol. 9, no. 3, august 2013.

[9] H. Berriri, W. Naouar, I. Bahri, I. Slama-Belkhodja, E. Monmasson, "Field programmable gate array-based fault-tolerant hysteresis current control for AC machine drives", IET Electric Power Applications,2011.

[10] N. Wada, K. Fujii, M.Saeki, "Reconfigurable fault-tolerant control of a vehicle with a steer-by-wire system”, International Conference on Advanced Mechatronic Systems (ICAMechS), 2012.

[11] Q. Jia; W. Chen; Y. Zhang; H. Li," Fault Reconstruction and Fault-Tolerant Control via Learning Observers in Takagi–Sugeno Fuzzy Descriptor Systems with Time Delays", IEEE Transactions on Industrial Electronics, pp 3885- 3895, 2015.

[12] T. Bekheira, R. Nassim, B. Mohamed, K. Abdelaziz, “A Control Reconfiguration Strategy for Post-Sensor FTC in Induction Motor-Based EVs,” IEEE Transaction on Vehicular Technology, Vol.62, No.3, pp. 965-971, 2013.

[13] M. Boussak, K. Jarray, “High-Performance Sensorless Indirect Stator Flux Orientation Control of Induction Motor Drive,” IEEE Transaction on Industrial Electronics, Vol.53, No.1, pp.14-49,2006.

[14] S. Bhoopendra, J. Shailendra, D. Sanjeet, “Direct Torque Control Induction Motor Drive with Improved Flux Response,”Hindawi Publishing Corporation Advances in Power Electronic, pp.1-11, 2012.

[15] A. Elbacha, Z. Boulghasoul, E. Elwarraki, “A Comparative Study of Rotor Time Constant Online Identification of an Induction Motor Using High Gain Observer and Fuzzy Compensator, WSEAS Transaction on System and Control, Vol.7, No.2, pp. 37-53,2012.

[16] P. Vojkan, M. Milutin, B. Nebojša, “Experimental verification of direct torque control methods for electric drive application,”Automatic Control and Robotic, Vol.8, pp. 111-126, 2009.

[17] S. Bhoopendra, J. Shailendra, D. Sanjeet, “Experimental Direct Torque Control Induction Motor Drive with Modified Flux Estimation and Speed control Algorithm,”IOSR Journal of Engineering, Vol.6, pp. 1296-1300, 2012.

[18] H. M. Hasanien, “FPGA implementation of Adaptative ANN controller for speed regulation of permanent stepper motor drives,” Energy Conversion and Management, pp.1252–1257, 2011.

[19] M. Dagbagi, L. Idkhajine, E. Monmasson, and I. Slama-Belkhodja, “FPGA Implementation of Power Electronic Converter Real-Time Model,”International Symposium on Power Electronics, Electrical Drives, Automation and Motion, pp. 658 – 663, 2012.

[20] E. Monmasson, I. Bahri, L. Idkhajine, A. Maalouf, and MW. Naouar, “Recent advancements in FPGA based controllers for AC drives application,” 13th international conference on optimization of electrical and electronic equipment, pp. 8-15, 2012.

[21] M. Shahbazi, P. Poure, S. Saadate, and MR. Zolghadri, “FPGA-Based Reconfigurable Control for Fault-Tolerant Back-to Back Converter without Redundancy, ” IEEE Transactions on Industrial Electronics, pp. 3360 – 3371, 2013

[22] K. Jezernik, J. Korelic, and R. Horvat, “PMSM Sliding Mode FPGA-Based Control for Torque Ripple Reduction,” IEEE Transactions on Power Electronic, pp. 3549 – 3556, 2013.

[23] Q.Jia, W.Chen, Y.Zhang, H. Li, "Fault Reconstruction and Fault-Tolerant Control via Learning Observers in Takagi–Sugeno Fuzzy Descriptor Systems With Time Delays", IEEE Transactions on industrial electronics, vol. 62, no. 6, June 2015.

[24] H. Berriri, W. Naoaur, I. Bahri, I. Slama-Belkhodja,E. Monmasson," Field programmable gate array-based fault-tolerant hysteresis current control for AC machine drives", IET Electric Power Applications 2011.

[25] J. Deepti Dasika, M.Saeedifard, "A Fault-Tolerant Strategy to Control the Matrix Converter under an Open-Switch Failure", IEEE transactions on industrial electronics, vol. 62, no. 2, February 2015.

[26] H.Guzman, F. Barrero, M.J. Duran,"IGBT-Gating Failure Effect on a Fault-Tolerant Predictive Current-Controlled Five-Phase Induction Motor Drive", IEEE transactions on industrial electronics, vol. 62, no. 1, January 2015.