Journal of Theoretical and Applied Information Technology
29th February 2012. Vol. 36 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

LOW-SPEED SENSORLESS CONTROL OF PMSM MOTOR DRIVE USING A NONLINEAR APPROACH BACKSTEPPING CONTROL: FPGA-BASED IMPLEMENTATION

[12]BADRE BOSSOUFI, [1]MOHAMMED KARIM, [2]SILVIU IONITA, [1]AHMED LAGRIOUI

[1)] STIC Team, Faculty of Sciences Dhar El Mahraz
University of Sidi Mohammed Ben Abdellah Fez, Morocco
[2)] Center of Modeling and simulation of the systems, Faculty of Electronics, Communications
University of PITEŞTI, Romania

E-mail: Badre_isai@hotmail.com, karim_lessi@yahoo.fr, silviu.ionita@upit.ro, lagrioui71@gmail.com

ABSTRACT

This paper presents a novel speed control technique for a permanent magnets synchronous (PMSM) drive based on newly nonlinear backstepping technique. We design a not adaptive speed regulator for a permanent-magnet synchronous motor (PMSM). In this paper, we present a new contribution of FPGAs (Field-Programmable Gate Array) for control of electrical machines. In this paper a robust continuous approach nonlinear not adaptative Control strategy for permanent-magnet synchronous motor (PMSM) drive systems is presented. This control scheme is based on an adaptive pole placement control strategy integrated to a Backstepping control scheme. The overall stability of the system is shown using Lyapunov technique. The simulation results clearly show that the proposed not adaptive scheme can track the speed reference. A bench test was realized by a prototyping platform, the experimental results obtained show the effectiveness and the benefit of our contribution and the different steps of implementation for the control FPGA.

Keywords: *Not Adaptive Backstepping control, Backstepping Design Technique, FPGAs, Permanent Magnet Synchronous Machine (PMSM), Lyaponov Stability, Robust Adaptive Control, FPGA-Based implementation.*

1. INTRODUCTION

The speed performance of new components and the flexibility inherent of all programmable solutions give today many opportunities in the field of digital implementation for control systems. This is true for software solutions as microprocessor or *DSP* (Digital Signal Processor). However, specific programmable hardware technology such as Field Programmable Gate Array (*FPGA*) can also be considered as an especially appropriate solution in order to boost performances of controllers [1-3]. Indeed, these generic components combine low cost development, thanks to their re- configurability, use of convenient software tools and more and more significant integration density [4, 5].

The *FPGA* technology is now used by an increasing number of designers in various fields of application such as signal processing [6], telecommunication, video, embedded control systems, and electrical control systems. This last domain, i.e. the studies of control of electrical machines, will be presented in this paper. Indeed, these components have already been used with success in many different applications such as Pulse Width Modulation (*PWM*), control of induction machine drives and multimachine system control. This is because the *FPGA-based* implementation of controllers can efficiently answer current and future challenges of this field.

This paper presents the realization of a platform for *not adaptative Backstepping* control of *PMSM* using *FPGA* based controller. This realization is especially aimed for future high performance applications. In this approach, not only the architecture corresponding to the control algorithm is studied, but also architecture and the *ADC* interface and *RS232 UART* architecture.

Among various types of ac motors, the permanent-magnet synchronous motor (PMSM) has received widespread acceptance in industrial applications due to some of its advantageous features, such as high efficiency, high torque-to-current ratio, low noise, and robustness [7]. For

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

high-performance control of a PMSM, a widely used approach is to employ the field-orientation mechanism [8], using synchronous frame proportional–integral (PI) current controllers with an active state-decoupling scheme. This technique allows a PMSM to achieve similar torque control performance to a separately excited dc motor where torque and flux can be controlled separately. This scheme, however, requires full knowledge of the machine parameters and operating conditions.

The PMSM motor is attractive candidates for high performance applications such as machine tools, or direct drive robotics. The effects of torque ripple on PMSM drives in such applications are potentially significant, as the high-frequency dynamics of the actuated system may get excited.

The PMSMs motors are known to provide higher torque per unit volume and better efficiency than induction motors, while improvements in the properties of permanent-magnet materials have increased their viability. Recently, sensorless PMSM drives have received increasing interest for industrial applications where there are limitations on the use of a position sensor. The elimination of the position sensor reduces the cost of the drive and increases the overall system ruggedness and reliability. High performance operation of sensorless PMSM drives mainly relies on accurate knowledge of the rotor-magnet flux magnitude, position, and speed. The sensorless rotor-position estimation techniques can be classified into two major groups: the motor model-based and the rotor-saliency-based techniques. The latter are suitable only for the interior PMSM (IPMSM). Most of the motor-model-based techniques detect the back-EMF vector, which holds information about the rotor position and speed, using either open-loop estimators [9] or closed-loop estimators/observers. In other motor-model-based techniques, the rotor-flux vector is directly estimated [10]. Moreover, adaptive observers have been used to estimate the stator current, the rotor speed, and the rotor position [11].

The adaptive backstepping design offers a choice of design tools for accommodation of uncertainties nonlinearities. And can avoid wasteful cancellations. In addition, the adaptive backstepping approach [12] is capable of keeping almost all the robustness properties of the mismatched uncertainties. The adaptive backstepping is a systematic and recursive design methodology for nonlinear feedback control.

The two major classes of controllers which are capable of dealing with nonlinear uncertain systems are adaptive and robust controllers. Backstepping control is an approach to nonlinear control design which has attracted a great deal of research interest in recent years. It is mainly applicable to systems having a cascaded or triangular structure.

The central idea of the approach is to recursively design controllers for motor torque constant uncertainty subsystems in the structure and ''step back'' the feedback signals towards the control input. This differs from the conventional feedback linearization in that it can avoid cancellation of useful nonlinearities in pursuing the objectives of stabilization and tracking. In addition, by utilizing the control Lyapunov function, it also has the flexibility in introducing appropriate dynamics to make the system behave in a desired manner.

The Backstepping control is a systematic and recursive design methodology for nonlinear feedback control. Appling those design methods, control objectives such as position, velocity can be achieved.

A nonlinear backstepping control design scheme is developed for the speed tracking control of PMSM that has exact model knowledge. The asymptotic stability of the resulting closed loop system is guaranteed according to Lyapunov stability theorem.

2. MODEL OF PMSM CONTROL SYSTEM

The model of IPMSM can be described in the *dq*-reference frame, with the d axis aligned with the direction of the rotor flux. So, the stator voltage can be characterized by [13],

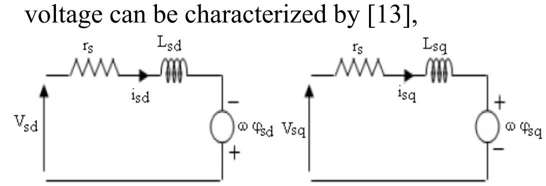

Fig.1: PMSM equivalent circuit from dynamic equations

$$\begin{bmatrix} V_{sd} \\ V_{sq} \end{bmatrix} = \begin{bmatrix} r_s & 0 \\ 0 & r_s \end{bmatrix} \begin{bmatrix} i_{sd} \\ i_{sq} \end{bmatrix} + \begin{bmatrix} p & \omega \\ \omega & p \end{bmatrix} \begin{bmatrix} \Phi_{sd} \\ \Phi_{sq} \end{bmatrix} \quad (1)$$

Where the direct and quadrate axis flux linkages are,

Journal of Theoretical and Applied Information Technology
29th February 2012. Vol. 36 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$$\Phi_{sd} = L_{sd}.i_{sd} + \Phi_f \quad (2)$$

$$\Phi_{sq} = L_{sq}.i_{sq} \quad (3)$$

The electromagnetic torque of the motor can be evaluated as follows,

$$Ce = \frac{3}{2} p\left[\phi_f.i_{sq} + (L_{sd} - L_{sq}).i_{sd}.i_{sq}\right] \quad (4)$$

The motor dynamics can be simply described by the equation (6).

$$Ce - C_r = J.\frac{d\Omega}{dt} + f.\Omega \quad (5)$$

With:
i_{sd} ,i_{sq} :Direct and quadrature axis stator currents
Ω: rotation's speed mechanical of the *PMSM.*
ω: rotation's speed electric.
p: Number of pairs of poles.
J : Total moment of inertia brought back on the tree of the *PMSM*.
f : Coefficient of viscous friction.
C_r: Resistive torque.
f: flux produced by the permanent magnet.
Φ_{sd}: d axis stator magnetic flux,
Φ_{sq}: q axis stator magnetic flux,
L_{sd}: d axis stator leakage inductance,
L_{sq}: q axis stator leakage inductance,
r_s: stator winding resistance,
C_e: electromagnetic torque,

3. NONLINEAR BACKSTEPPING NOT ADAPTATIVE CONTROLLER APPLIED TO PMSM

No Adaptive backstepping control is a control technique that can effectively linearize a nonlinear system such as the PMSM in the presence of uncertainties. Unlike other feedback linearization techniques, adaptive backstepping has the flexibility of keeping useful non linearities intact during stabilization. The essence of backstepping is the stabilization of a virtual control state. Hence, it generates a corresponding error variable which can be stabilized by carefully selecting proper control inputs. These inputs can be determined from Lyapunov stability analysis.

From (1), it is obvious that the dynamic model of PMSM is highly nonlinear because of the coupling between the speed and the electrical currents. According to the vector control principle, the direct axis current i_d is always forced to be zero in order to orient all the linkage flux in the d axis and achieve maximum torque per ampere.

$$\begin{cases}\frac{di_{sd}}{dt} = -\frac{r_s}{L_{sd}}.i_{sd} + \frac{L_{sq}}{L_{sd}} p\Omega i_{sq} + \frac{V_{sd}}{L_{sd}} \\ \frac{di_{sq}}{dt} = -\frac{r_s}{L_{sq}}.i_{sq} - \frac{L_{sd}}{L_{sq}} p\Omega i_{sd} - \frac{\Phi_f}{L_{sq}} p\Omega + \frac{V_{sq}}{L_{sq}} \\ \frac{d\Omega}{dt} = \frac{3p}{2J}(\Phi_f i_{sq} + (L_{sd} - L_{sq}) i_{sd} i_{sq}) - \frac{f}{J}\Omega + \frac{C_r}{J}\end{cases} \quad (6)$$

The vector $[x] = [x_1 \quad x_2 \quad x_3]^T = [i_{sd} \quad i_{sq} \quad \Omega]^T$ choice as state vector is justified by the fact that currents and speed are measurable and that the control of the instantaneous torque can be done comfortable via the currents i_{sd} and/or i_{sq}. And stator voltages as control variables $u = [V_{sd} \quad V_{sq}]^T$.

The main objective of the backstepping control is to regulate the speed of the machine to its reference value Ω_{ref} whatever external disturbances.

It is assumed that the engine parameters are known and invariant.

By choosing $[i_{sd} \quad i_{sq} \quad \Omega]^T$ as variable states and equation (6) the mathematical model of the machine. The objective is to regulate the speed to its reference value.

3.1. Backstepping speed controller

We begin by defining the tracking errors:

$$e_\Omega = \Omega_{ref} - \Omega \quad (7)$$

The derivative of (7) is:

$$\begin{aligned}\dot{e}_\Omega &= \frac{de_\Omega}{dt} = \dot{\Omega}_{ref} - \dot{\Omega} \\ &= \dot{\Omega}_{ref} - \frac{1}{J}\left[\frac{3p}{2}(\Phi_f i_{sq} + (L_{sd} - L_{sq}) i_{sd} i_{sq}) - f\Omega - C_r\right]\end{aligned} \quad (8)$$

We define the following quadratic function:

$$V_1 = \frac{1}{2} e_\Omega^2 \quad (9)$$

Its derivative along the solution of (9), is given by:

$$\begin{aligned}\dot{V}_1 &= e_\Omega \dot{e}_\Omega \\ &= e_\Omega\left(\dot{\Omega}_{ref} - \frac{1}{J}\left[\frac{3p}{2}(\Phi_f i_{sq} + (L_{sd} - L_{sq}) i_{sd} i_{sq}) - f\Omega - C_r\right]\right)\end{aligned} \quad (10)$$

Using the backstepping design method, we consider the d-q axes currents components i_{sd} and i_{sq} as our virtual control elements and specify its desired behavior, which are called stabilizing function in the backstepping design terminology as follows:

Journal of Theoretical and Applied Information Technology
29th February 2012. Vol. 36 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$$i_{sdref} = 0$$
$$i_{sqref} = \frac{2}{3p\Phi_f}(f\Omega + C_r + J.k_\Omega.e_\Omega) \qquad (11)$$

Where k_Ω is a positive constant

Substituting (11) in (10) the derivative of V_1:

$$\dot{V}_1 = -k_\Omega e_\Omega^2 \leq 0 \qquad (12)$$

3.2. Backstepping Current controller

We have the asymptotic stability of the origin of the system (6). We defined:

$$\begin{cases} e_d = i_{sdref} - i_{sd} \quad with \quad i_{sdref} = 0 \\ e_q = i_{sqref} - i_{sq} \end{cases} \qquad (13)$$

Their dynamics can be written:

$$\dot{e}_d = \dot{i}_{sdref} - \dot{i}_{sd} = \frac{r_s}{L_{sd}}.i_{sd} - \frac{L_{sq}}{L_{sd}}p\Omega i_{sq} - \frac{V_{sd}}{L_{sd}}$$
$$\dot{e}_q = \dot{i}_{sqref} - \dot{i}_{sq} = \frac{2}{3p\Phi_f}(f\dot{\Omega} + C_r + J.k_\Omega.\dot{e}_\Omega) + \qquad (14)$$
$$+\frac{r_s}{L_{sq}}.i_{sq} + \frac{L_{sd}}{L_{sq}}p\Omega i_{sd} + \frac{p\Phi_f}{L_{sq}}\Omega - \frac{V_{sd}}{L_{sd}}$$

To analyze the stability of this system we propose the following Lyapunov function:

$$V_2 = \frac{1}{2}(e_\Omega^2 + e_d^2 + e_q^2) \qquad (15)$$

Its derivative along the trajectories (12), (13) and (14) is:

$$\dot{V}_2 = \frac{1}{2}(e_\Omega \dot{e}_\Omega + e_d \dot{e}_d + e_q \dot{e}_q)$$
$$= -k_\Omega e_\Omega^2 - k_d e_d^2 - k_q e_q^2 +$$
$$e_d[k_d e_d - \frac{V_{sd}}{L_{sd}} + \frac{r_s}{L_{sd}} - \frac{L_{sq}}{L_{sd}}\Omega i_{sq} +$$
$$\frac{3p}{2J}(L_{sd} - L_{sq})e_\Omega i_{sq}] + e_q[k_q e_q + \frac{2(k_\Omega J - f)}{3p\Phi_f}.$$
$$(\frac{3p\Phi_f}{2J}e_q + \frac{3p}{2J}(L_{sd} - L_{sq})e_d i_{sq} - k_\Omega e_\Omega) \qquad (16)$$
$$+\frac{3p\Phi_f}{2J}e_\Omega - \frac{V_{sq}}{L_{sq}} + \frac{r_s}{L_{sq}}i_{sq} + \frac{L_{sd}}{L_{sq}}\Omega i_{sd} + \Omega\frac{\Phi_f}{L_{sq}}]$$

The expression (16) found above requires the following control laws:

$$V_{sd} = k_d L_{sd} e_d + r_s i_{sd} - L_{sq}\Omega i_{sq}$$
$$+\frac{3pL_{sd}}{2J}(L_{sd} - L_{sq})e_\Omega i_{sq}$$
$$V_{sq} = \frac{2L_{sq}(k_\Omega J - f)}{3p\Phi_f}\left(\frac{3p\Phi_f}{2J}e_q + \frac{3p}{2J}(L_{sd} - L_{sq})e_d i_{sq} - k_\Omega e_\Omega\right) \qquad (17)$$
$$+\frac{3p\Phi_f L_{sq}}{2J}e_\Omega + r_s i_{sq} + L_{sd}\Omega i_{sd} + \Omega\Phi_f + k_q L_{sq} e_q$$

With this choice the derivatives of (17) become:

$$\dot{V}_2 = -k_\Omega e_\Omega - k_d e_d - k_q e_q \leq 0 \qquad (18)$$

3.3. Simulation results

In order to verify asymptotic stability of not-adaptive controller and show its performance against and tracking the trajectory of the speed, we implemented the system in Matlab / Simulink environment as the diagram below:

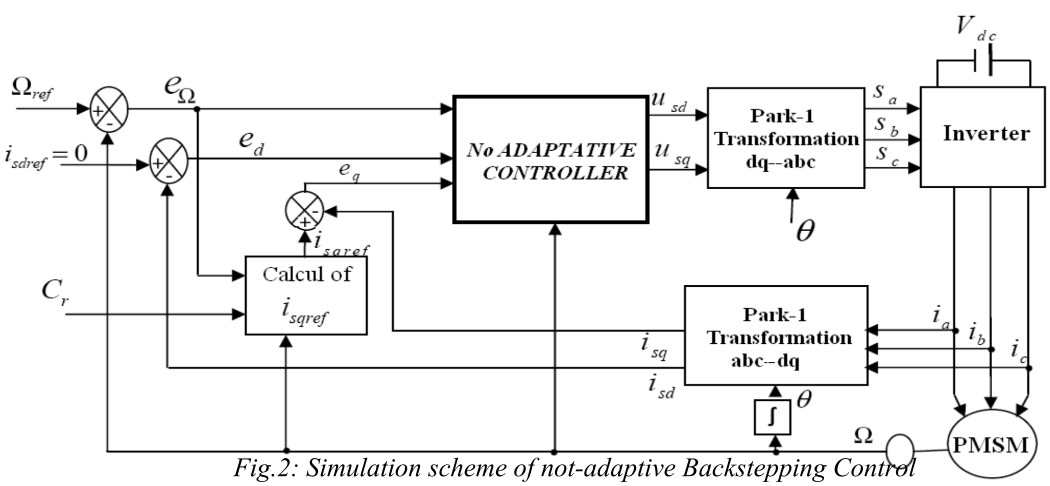

Fig.2: Simulation scheme of not-adaptive Backstepping Control

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Next, we performed simulations as follows:

- For test the further trajectory, the speed reference was made variable. The reference of the direct component of current was set to zero.
- With the aim to test the performance of the controller in disturbance rejection, we apply a load torque 7N.m at t = 0.6s.
- To test the robustness of the controller, electrical and mechanical parameters are varied in the non-adaptive controller at t=0.6s. Indeed, the values of the resistance and inductance are directly increased by 50% of the nominal value while the value of inductance in quadrature becomes equal to half of its nominal value. On the other hand, the value of the magnetic flux is increased by 20% of its nominal value.

For a trajectory Ω_{ref}=150 rad/s at 0s, i_{sdref}=4.5A and C_r=7N.m.the following figures (3 to 6) show the performance of the input output linearization control.

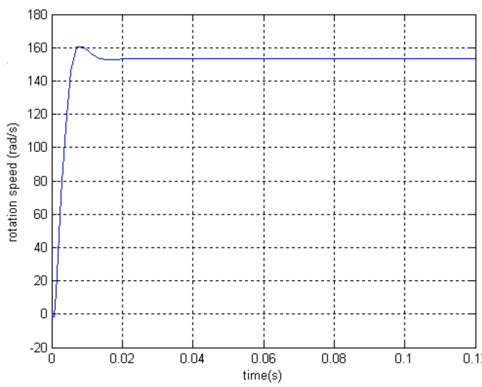

Fig.3 : Speed response trajectory

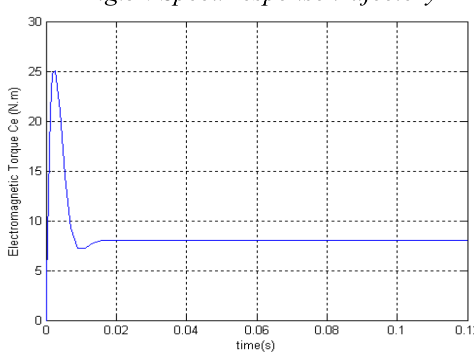

Fig.4: Electromagnetic Torque Ce reponse

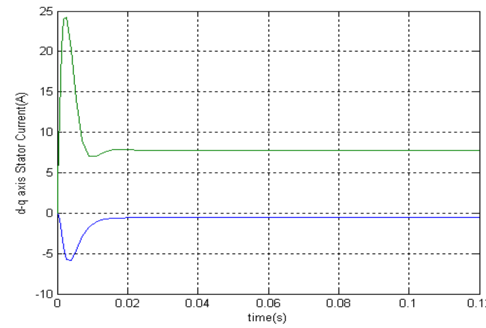

Fig.5: d-q axis current without uncertainties

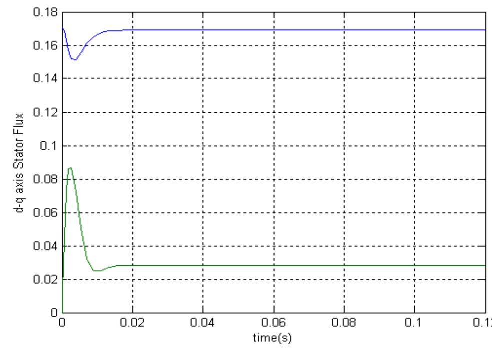

Fig.6: d-q axis Flux without uncertainties

- **Further trajectory**

For a trajectory Ω_{ref} (=150rad/s at 0s, =100rad/s at t=0.4s, =250rad/s at t=0.8s), the following figures (7 to 12) show the performance of the PMSM.

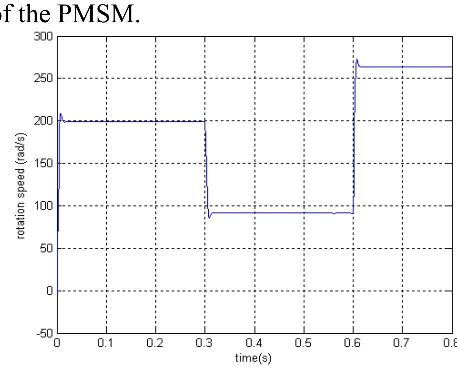

Fig.7: Speed response trajectory

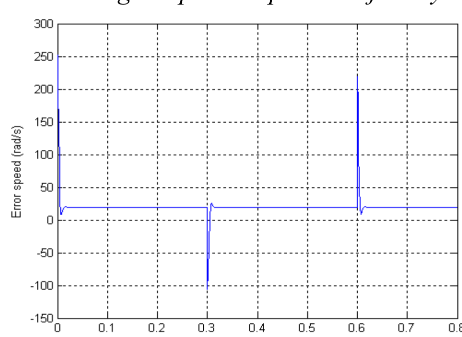

Fig.8: Error Speed response

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

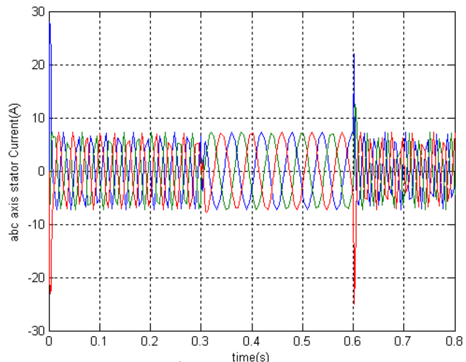

Fig.9: abc Axis Stator Current

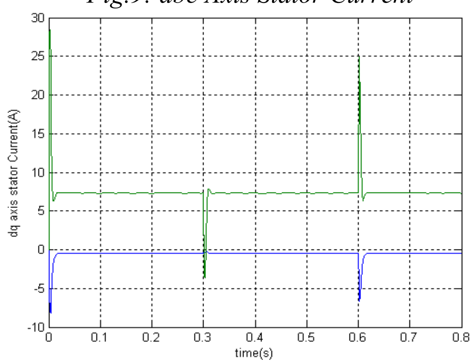

Fig.10: dq Axis Stator Current

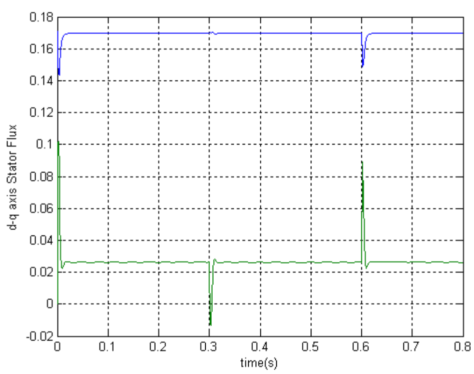

Fig.11: dq Axis Stator Flux

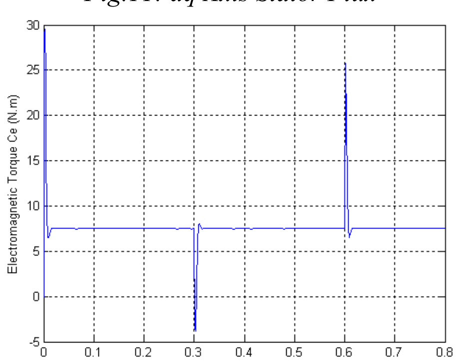

Fig.12: Electromagnetic Torque Ce

- Rejection of disturbance load torque

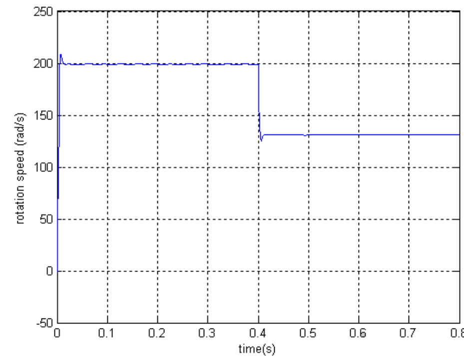

Fig.13: Speed response

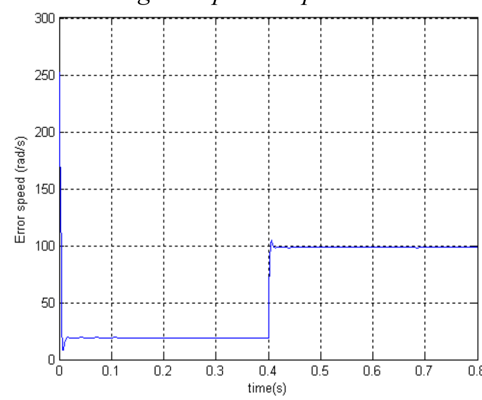

Fig.14: Error Speed response

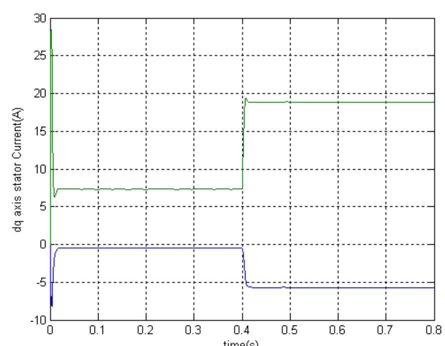

Fig.15: dq Axis Stator Current

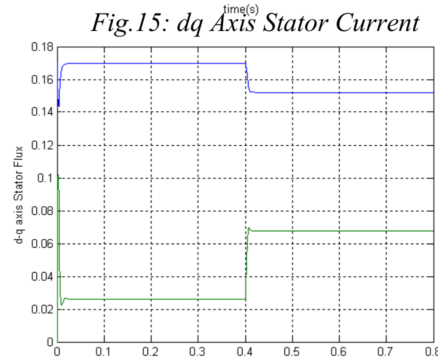

Fig.16: dq Axis Stator Flux

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

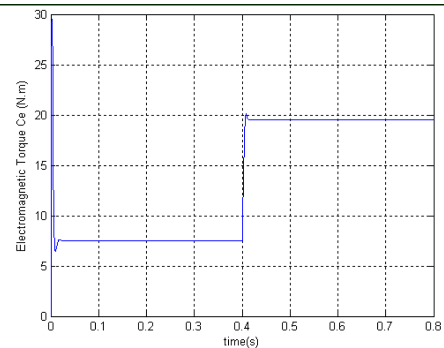

Fig.17: Electromagnetic Torque Ce

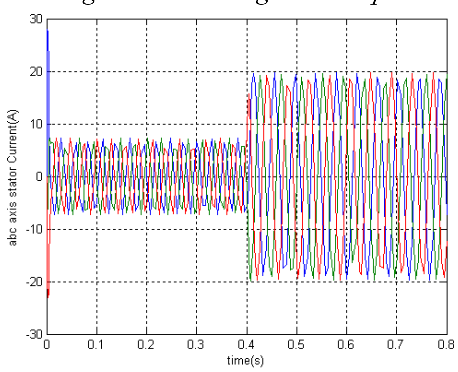

Fig.18: abc Axis Stator Current

The simulation results show the robustness of the controller used. In the third part we will define in brief the different steps for the implementation of an algorithm on the FPGA target. Also explained the platform we used to make this work.

4. FPGA IMPLEMENTATION OF AN ROBUST NO ADAPTIVE BACKSTEPPING CONTROLLER

4.1. Development Of The Implementation

There are several manufacturers of *FPGA* components such: Actel, Xilinx and Altera…etc. These manufacturers use different technologies for the implementation of *FPGAs*. These technologies are attractive because they provide reconfigurable structure that is the most interesting because they allow great flexibility in design. Nowadays, *FPGAs* offer the possibility to use dedicated blocks such as *RAMs*, multipliers wired interfaces *PCI* and *CPU* cores.

The architecture designing was done using with *CAD* tools. The description is made graphically or via a hardware description language high level, also called *HDL* (Hardware Description Language). Is commonly used language *VHD*L and Verilog. These two languages are standardized and provide the description with different levels, and especially the advantage of being portable and compatible with all *FPGA* technologies previously introduced [14-18].

The *Fig.19* summarizes the different steps of programming an FPGA. The synthesizer generated with *CAD* tools first one Netlist which describes the connectivity of the architecture. Then the placement-routing optimally place components and performs all the routing between different logic. These two steps are used to generate a configuration file to be downloaded into the memory of the *FPGA*. This file is called bitstream. It can be directly loaded into *FPGA* from a host computer.

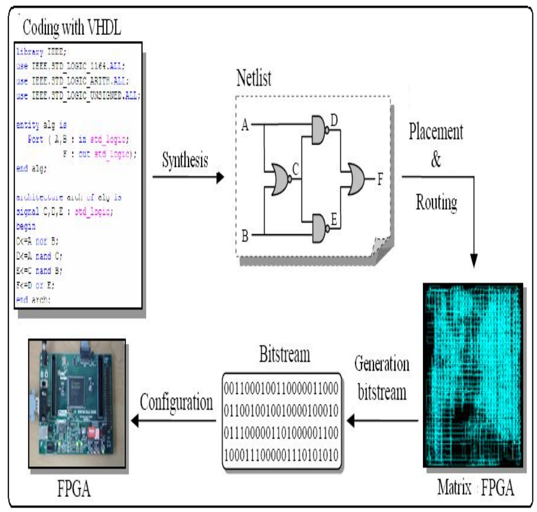

Fig.19: Programming FPGA devisees

In this paper an *FPGA XC3S500E* Spartan3E from Xilinx is used. This *FPGA* contains 400,000 logic gates and includes an internal oscillator which issuer a 50MHz frequency clock. The map is composed from a matrix of 5376 slices linked together by programmable connections.

4.2. Simulation Procedure

The simulation procedure begins by verifying the functionality of the control algorithm by trailding a functional model using Simulink's System Generator for Xilinx blocks. For this application, the functional model consists in a Simulink timeis discretired model of the *No adaptative Backstepping* algorithm associated with a voltage inverter and PMSM model. The *Fig.20* shows in detail the programming of the control shown in *Fig.2* in the S*YSTEM GENERATOR* environment from Xilinx, we will implement it later in the memory of the *FPGA* for the simulation of *PMSM*.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

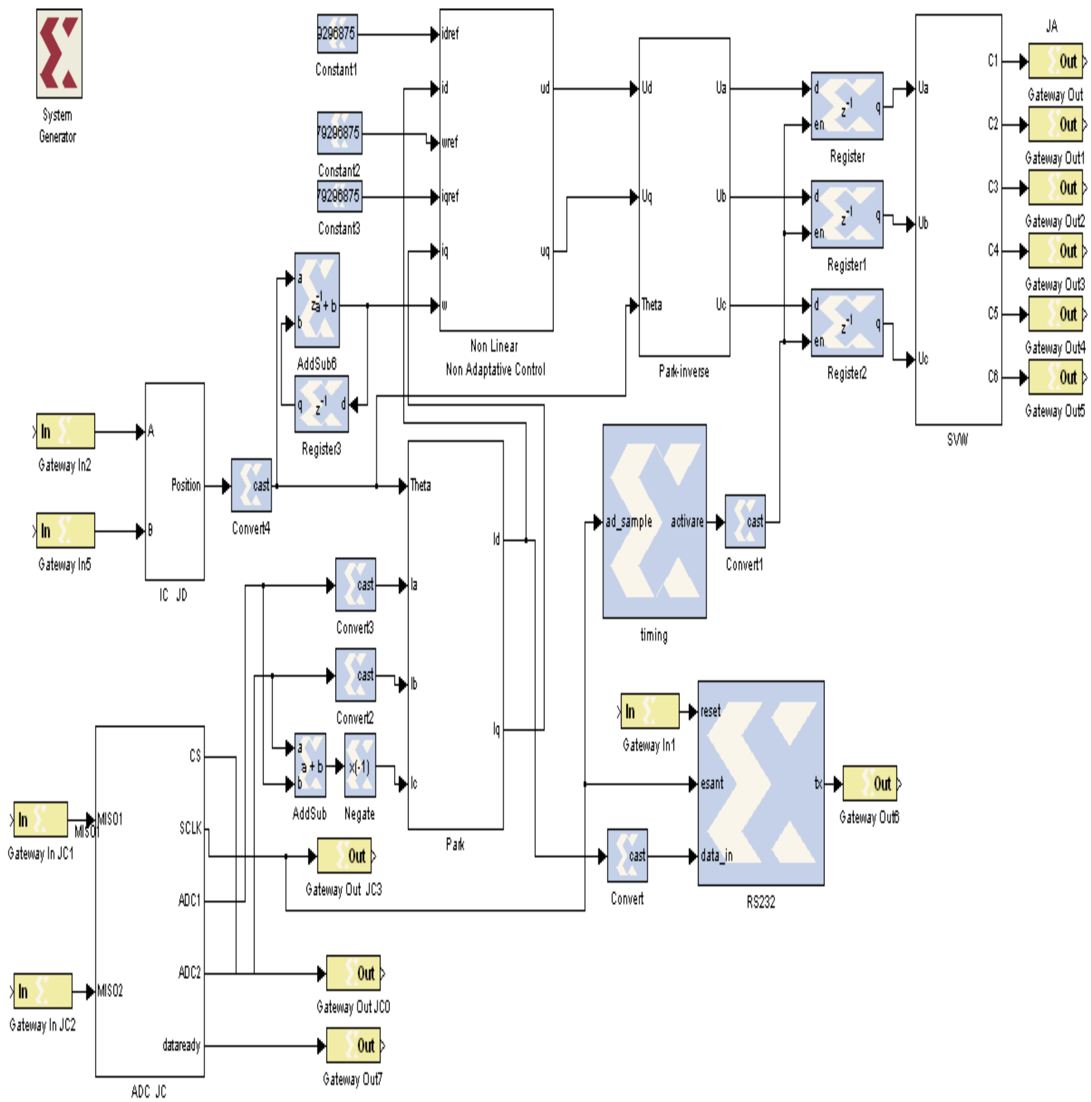

Fig.20: Functional Model for No adaptative Backstepping Controller from SYSTEM GENERATOR

The functional model for no adaptative Backstepping Controller from SYSTEM GENERATOR is composed the different blocks:

- The block encoder interface *IC* allows the adaptation between the *FPGA* and the acquisition board to iniquity the rotor position of the *PMSM*,
- The *ADC* interface allows the connection between the *FPGA* and the analog-digital converter (*ADCS7476MSPS 12-bit A / D*) that will be bound by the following two Hall Effect transducers for the acquisition of the stator currents machine,
- The blocks of coordinate's transformation: the transformation of Park Inverse (*abc-to-dq*),
- The blocks of coordinate's transformation: the transformation of Park (*dq-to-abc*),
- The *SVW* block is the most important, because can provide control pulses to the *IGBT* voltage inverter in the power section from well-regulated voltages,
- The block for the controller no adaptative backstepping which is designed to regulate the speed and stator currents of the PMSM,
- Block "Timing" which controls the beginning and the end of each block, which allows the refresh in the voltages reference V_{10}, V_{20} and V_{30} at the beginning of each sampling period,
- The *RS232* block allows signal timing and recovery of signals viewed, created by another program on Matlab & Simulink to visualize the desired output signal.

The second step of the simulation is the determination of the suitable sampling period and

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

fixed point format. The *Fig.21* gives the specification model of *abc-to-dq* (park) transformation.

For example, we present the construction of Block Clark's Transformation in the system generator environment from Xilinx, which is characterized by the following system (18):

$$\begin{bmatrix} V_{sd} \\ V_{sq} \\ V_0 \end{bmatrix} = \sqrt{\frac{2}{3}} \begin{bmatrix} Cos\theta & Cos(\theta - \frac{2\pi}{3}) & Cos(\theta - \frac{4\pi}{3}) \\ -Sin\theta & -Sin(\theta - \frac{4\pi}{3}) & -Sin(\theta - \frac{4\pi}{3}) \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} V_{sa} \\ V_{sb} \\ V_{qc} \end{bmatrix} \quad (18)$$

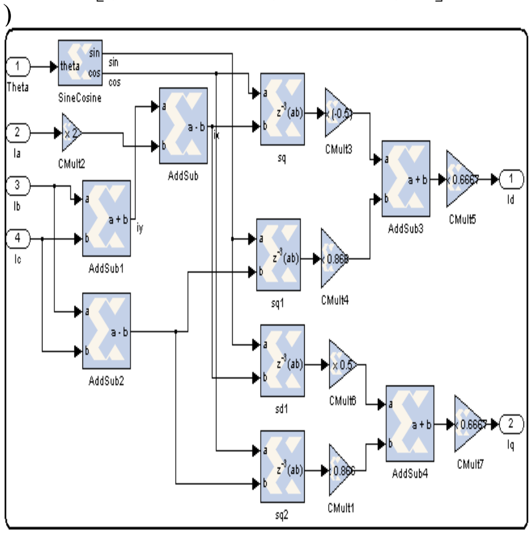

Fig.21: Park Transformation Model.

The specification model is then used for the definition of the corresponding Data Flow Graph. The *Fig.22* shows the *DFG* corresponding to the Park transformation module.

4.3. Modular design of the control architecture

To design the control architecture to implement a datapath and control unit is defined for each module as shown in *Fig.22*. The datapath is mainly made up of operators and registers. The registers can provide the technical Pipeline which consists on decomposing the performance of an algorithm in different sequences according to the data dependency. The communication between the registers and operators can be achieved through data buses, multiplexers and demultiplexers. Concerning the control unit, its role is to ensure the flow of data according to very specific sequences within the datapath [19-20].

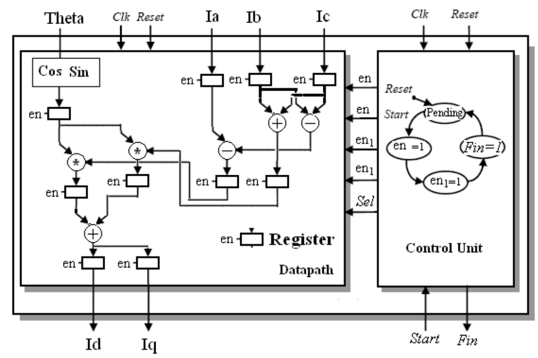

Fig.22: DFG Clark transformation module architecture

5. EXPERIMENTAL RESULTS

5.1. Simulation of the no adaptative backstepping control

For this work, the used *FPGA* target is a *XC3S500E Spartan3E* the firm Xilinx, the *FPGA* based hardware control system includes the *no adaptative backstepping control*, an *ADC* interface and *IC* interface in one *FPGA* chip. *Fig.25* presents the corresponding implemented architecture.

The control unit for architecture ensures implanted control module of the *ADC* interface, the encoder interface and the command encoder *DTC*. At the beginning of each sampling period, the *A/D* interface module and encoder interface are activated simultaneously. Then, after a delay conversion from analog to digital, the control module *Backstepping Control* is activated. This is driven by its own control unit and allows you to refresh the reference d-axis stator current and q-axis stator courant. Once activated, the control unit control module *Backstepping Control* activates the first module of the transformation of Park *(abc-dq)* that will calculate the components i_{sd} and i_{sq} stator current vector and the speed calculation for the electrical machine. When the module of this transformation indicates the end of the calculation, the "*No Adaptative Backstepping Controller*" module is activated.

Subsequently, when "*No Adaptative Backstepping Controller*" module indicates the end of the calculation, the module of the transformation of Park inverse *(dq-abc)* is activated to calculate the components V_{sa}, V_{sb} and V_{sc} stator voltage vector. Finally, the module of the *PMW* is activated. The latter has a running time and generates the control signals S_a, S_b and S_c for controlling the switches of the voltage inverter.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

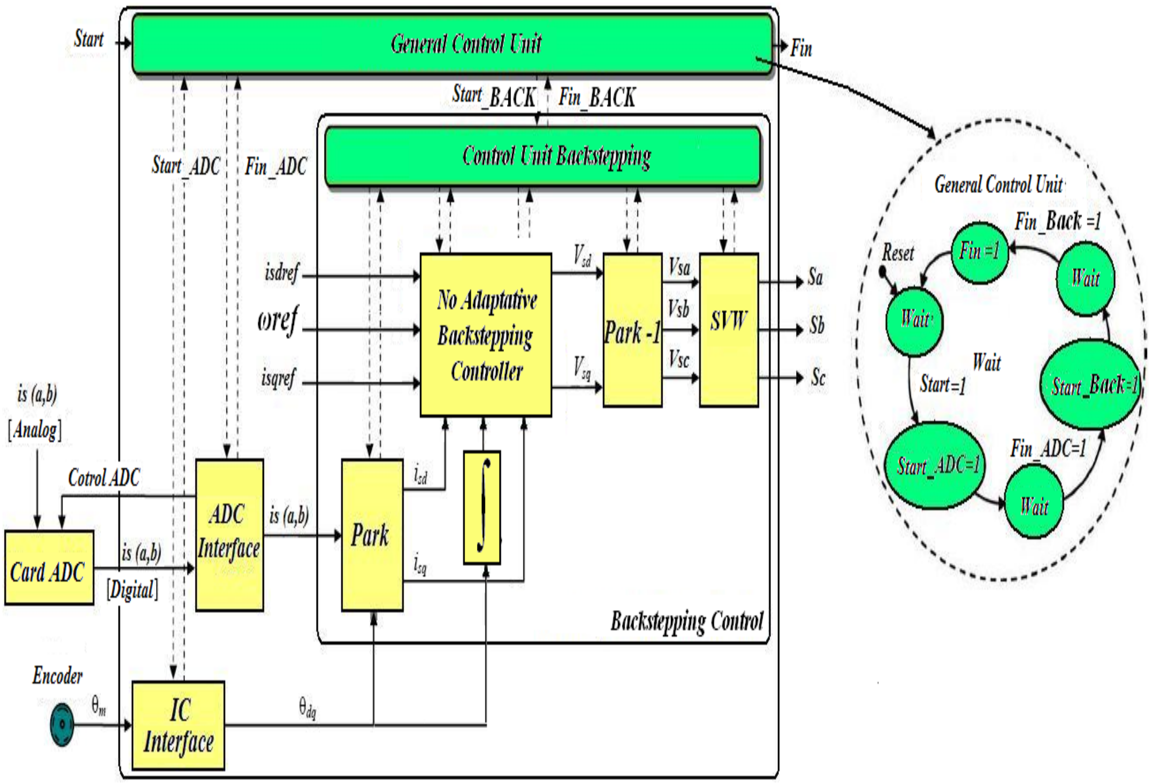

Fig.23: FPGA based hardware No Adaptative Backsteppin Control

5.2. Prototyping platform

To test the *FPGA* based controller, a prototyping platform for the control of a Permanent magnet Synchronous Machine was assembled (*Fig.24*).

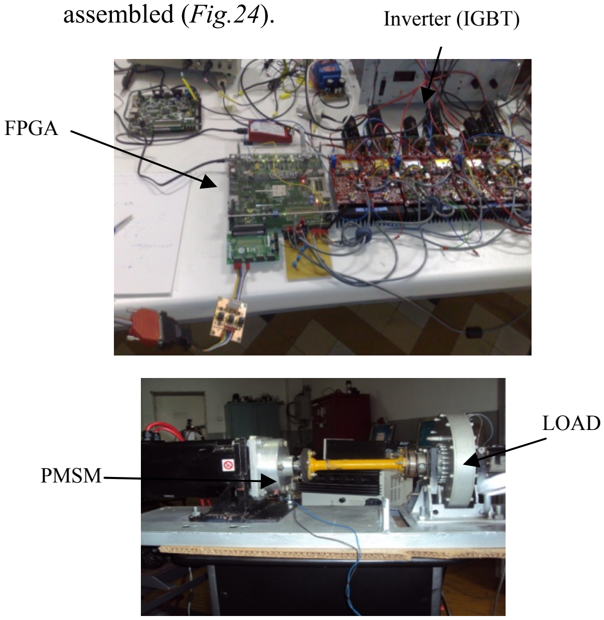

Fig.24: Prototyping platform control

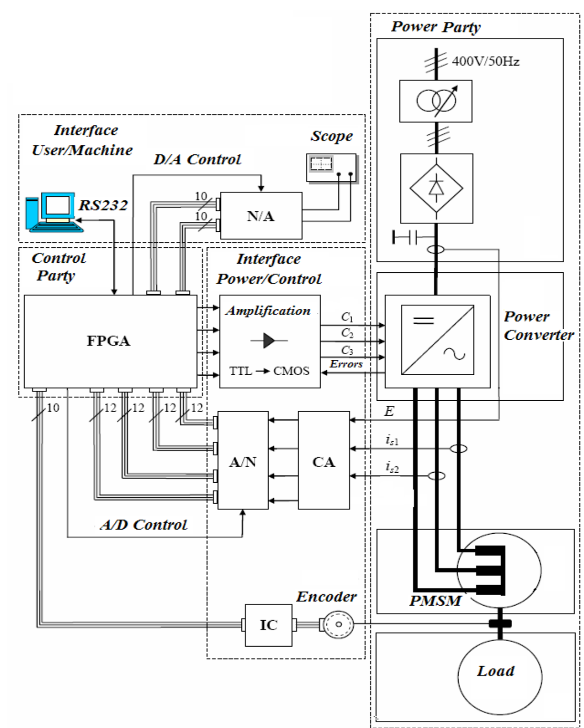

Fig.25: Synoptic Prototyping platform control

The *Fig.25* shows the block diagram of the implementation of the platform to ban testing for the simulation of PMSM controlled by an FPGA. It shows the different parts, such as the power section and control section and the man/machine interface and the procedure for the acquisition of current and voltage of the DC bus.

5.3. Simulation Results

In *Fig.26* the experimental results No Adaptative Backstepping Control of *PMSM* with the *FPGA* platform are shows the evolution of the stator current i_{sd} which shows that the output follows the reference i_{sdref} and i_{sq}. The *Fig.27* shows the stator current i_{sa} and i_{sb}. Update frequency for this implementation is 20 kHz.

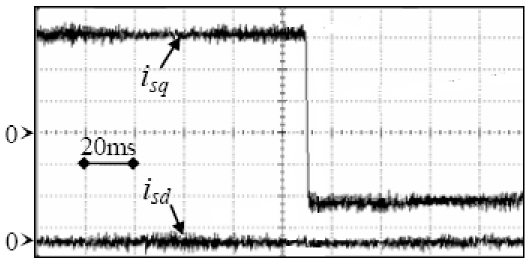

Fig.26: d-q axis stator current performances

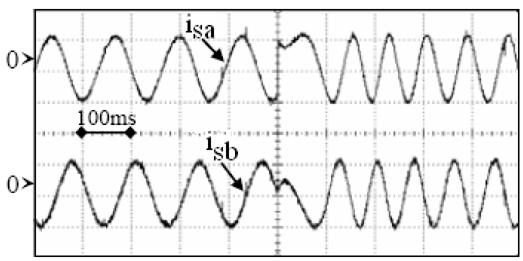

Fig.27: abc axis stator current performances

6. CONCLUSION

In this paper a robust continuous approach nonlinear not adaptative Control strategy for permanent-magnet synchronous motor (PMSM) drive systems is presented. The FPGA based implementation is detailed, a bench test was realized by a prototyping platform, the experimental results obtained show the effectiveness and the benefit of our contribution and the different steps of implementation for the control FPGA. A method of nonlinear backstepping control has been proposed and used for the control of a PMSM. The simulation results show, with a good choice of control parameters, good performance obtained with the proposed control as compared with the non-linear control by state feedback. This shows the robustness and reliability of the control algorithm used.

ACKNOWLEDGEMENTS

We thank all those who contributed to make this work, including my teachers in my home laboratory STIC Team, Faculty of Science Dhar El Mahraz, Fez, Morocco and the *Center of Modeling and simulation of the systems,* University of Pitesti, Romania, my wife and all my friends for their support.

This work was performed in the Montefiore Institute, University of Liege (ULG) in Belgium, mobility funded by AVERROES (Erasmus) Program in the framework of the European Erasmus Mundus Action 2.

REFERENCES:

[1]. Y.Y. Tzou, H.J. Hsu: “FPGA Realization of Space-Vector PWM Control IC for Three Phase PWM Inverters,” IEEE Trans, Power Electron., vol.12, n°6, pp.953-963, nov.1997.

[2].Melicio, R; Mendes, VMF; Catalao, JPS “Modeling, Control and Simulation of Full-Power Converter Wind Turbines Equipped with Permanent Magnet Synchronous Generator” vol 5 (issue 2) pg 397-408 Mar - Apr 2010.

[3].J. J. Chen and K. P. Chin, "Automatic flux-weakening control of permanent magnet synchronous motors using a reduced-order controller," IEEE Trans. Power Electron., vol. 15, pp. 881-890, Sept. 2000.

[4].M. Rodic, K. Jezernik, “ Speed Sensoless Sliding Mode Torque Control of Induction Motor “, *IEEE Trans on. Industrial Electronics,* February 2002.

[5].Hisn-Jang Shieh and Kuo-Kai Shyu, “Non Linear Sliding Mode Torque Control With Adaptive Backstepping Approach For Induction Motor Drive”, *IEEE Transactions on Industrial Electronics*, Vol. 46, N° 2, April 1999, pp. 380-388.

[6].Z. P. Jiang and H. Nijmeijer, “Tracking control of mobile robots: A case study in backstepping,” *Automatica*, vol. 33, no. 7, pp. 1393–1399, Jul. 1997.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[7].T. C. Lee, K. T. Song, C. H. Lee, and C. C. Teng, "Tracking control of unicycle-modeled mobile robots using a saturation feedback controller," *IEEE Trans. Control Syst. Technol.*, vol. 9, no. 2, pp. 305–318, Mar. 2001.

[8].T. H. Li, S. J. Chang, and Y. X. Chen, "Implementation of humanlike driving skills by autonomous fuzzy behavior control on an FPGA based car-like mobile robot," *IEEE Trans. Ind. Electron.*, vol. 50, no. 5, pp. 867–880, Oct. 2003.

[9].S. S. Solano, A. J. Cabrera, I. Baturone, F. J. Moreno-Velo, and M. Brox, "FPGA implementation of embedded fuzzy controllers for robotic applications," *IEEE Trans. Ind. Electron.*, vol. 54, no. 4, pp. 1937– 1945, Aug. 2007.

[10]. Y. F. Chan, M. Moallem, and W. Wang, "Design and implementation of modular FPGA-based PID controllers," *IEEE Trans. Ind. Electron.*, vol. 54, no. 4, pp. 1898–1906, Aug. 2007.

[11]. S. H. Han, M. H. Lee, and R. R. Mohler, "Real-time implementation of a robust adaptive controller for a robotic manipulator based on digital signal processors," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 29, no. 2, pp. 194–204, Mar. 1999.

[12]. D. Zhang and H. Li, "A stochastic-based FPGA controller for an induction motor drive with integrated neural network algorithms," *IEEE Trans. Ind. Electron.*, vol. 55, no. 2, pp. 551–561, Feb. 2008.

[13]. C. F. Jung and C. H. Hsu, "Temperature control by chip implemented adaptive recurrent fuzzy controller designed by evolutionary algorithm," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 52, no. 11, pp. 2376–2384, Nov. 2005.

[14]. C. F. Jung and J. S. Chen, "Water bath temperature control by a recurrent fuzzy controller and its FPGA implementation," *IEEE Trans. Ind. Electron.*, vol. 53, no. 3, pp. 941–949, Jun. 2006.

[15]. B.Bossoufi, M.Karim, S.Ionita, A.Lagrioui, "DTC CONTROL BASED ARTIFICIAL NEURAL NETWORK FOR HIGH PERFORMANCE PMSM DRIVE" *Journal of Theoretical and Applied Information Technology JATIT*, pp165-176, Vol. 33 No.2, 30th November 2011.

[16]. B.Bossoufi, M.Karim, S.Ionita, A.Lagrioui, G.Iana "Matlab & Simulink Simulation with FPGA-Based Implementation Sliding Mode Control of a Permanent Magnet Synchronous Machine Drive." *WSEAS TRANSACTIONS on SYSTEMS and CONTROL*, pp 92-103, Issue 3 Vol.6, March 2011.

[17]. B.Bossoufi, M.Karim, S.Ionita, A.Lagrioui, "Indirect Sliding Mode Control of a Permanent Magnet Synchronous Machine: FPGA-Based Implementation with Matlab & Simulink Simulation" *Journal of Theoretical and Applied Information Technology JATIT*, pp32-42, Vol. 29 No.1, 15th July 2011.

[18]. B.Bossoufi, M.Karim, S.Ionita, A.Lagrioui, "The Optimal Direct Torque Control of a PMSM drive: FPGA-Based Implementation with Matlab & Simulink Simulation" *Journal of Theoretical and Applied Information Technology JATIT*, pp63-72, Vol. 28 No.2, 30th June 2011.

[19]. B.Bossoufi, M.Karim, S.Ionita, A.Lagrioui, "Performance Analysis of the Direct Torque Control and the Space Vector Modulation for Permanent Magnet Synchronous Motor Drive" *University of Pitesti – SCIENTIFIC BULLETIN (Journal)*, Series Electronic and Computer Science, pp 23-34, December 2010.

[20]. A. Lagrioui, H. Mahmoudi, B.Bossoufi, "DISCRETE LINEAR PREDICTIVE CONTROL OF PERMANENT MAGNET SYNCHRONOUS MOTOR (PMSM)" *Journal of Theoretical and Applied Information Technology JATIT*, pp21-28, Vol. 31 No.1, 15th September 2011.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

AUTHOR PROFILE:

Badre BOSSOUFI Was born in Morocco in 1985, He received the M.S degree in electrical engineering (Automated Industrial Systems Engineering ISAI) from faculty of Science Dhar El Mahraz, Sidi Mohammed Ben Abdellah University in 2009, Fez, Morocco. He is currently working toward the Ph.D. degree in electrical engineering in the STIC Team from FSDM Fez and Center of Modeling and simulation of the systems, Faculty of Electronics, Communications, University of PITEŞTI, Romania. Member of Montefiore Institute, Department Electrical, University of Liege, Belgium.

His research interests are in the areas of power electronics and control (DTC, FOC, Sliding Mode, Backstepping control…), which includes ac machine drives and FPGA-based controllers, and use of artificial intelligence techniques for the control of electrical machines.

He's a graduate student Member of Industrial Electronics, Electronics Devices, Power Electronics Society of the Institute of Electrical and Electronics Engineers IEEE.