FPGA-Based Implementation Sliding Mode Control and nonlinear Adaptive backstepping control of a Permanent Magnet Synchronous Machine Drive

BADRE BOSSOUFI [1)2)], MOHAMMED KARIM [1)], AHMED LAGRIOUI[1)],
[1)] Laboratory of Electrical Engineering and Maintenance.
Higher School of Technology-Oujda, University of Mohammed I, Morocco.
[2)] STIC Team, Faculty of Sciences Dhar El Mahraz Fez, Morocco
Badre_isai@hotmail.com

Abstract: In this paper, the adaptive and non-adaptive nonlinear Backstepping control approach for a permanent magnet synchronous motor drive is discussed and analyzed. We present a Matlab&Simulink simulation and experimental results from a benchmark based on FPGA. The design of a controller for a nonlinear system where the state vector dimension is high, can often be a difficult task, if not impossible. The Backstepping technique provides a systematic method to address this type of problem. It combines the notion of Lyapunov function and a controller procedure recursively. First, the adaptive and no adaptive Backstepping control approach is utilized to obtain the robustness for mismatched parameter uncertainties. The overall stability of the system is shown using Lyapunov technique. The simulation results clearly show that the proposed scheme can track the speed reference. Secondly, some experimental results are demonstrated to validate the proposed controllers. The experimental results carried from a prototyping platform are given to illustrate the efficiency and the benefits of the proposed approach and the various stages of implementation of this structure in FPGA.

Keywords: Not Adaptive Backstepping control; Backstepping Design Technique; FPGAs; Permanent Magnet Synchronous Machine (PMSM); Lyapunov Stability; Robust Adaptive Control; Systems Generator; Reusability.

1. Introduction

Three-phase Permanent Magnet Synchronous Motor (PMSMs) is strongly used in industry and consumes more than 70% of industrial electricity. This is why considerable efforts and different searches are being done to improve their performances and their efficiency. The efficiency of electrical machine drives is greatly reduced at light loads, where the flux magnitude reference is held on its initial value. The loss minimization is realized using high-quality materials and excellent design procedures in the manufacturing process. Moreover, expert control algorithms are employed in order to improve machine performance. In this paper we are interested in two mode controls for PMSM drive, the not adaptive backstepping and indirect sliding mode control (ISMO).

The not adaptive backstepping approach offers a choice of design tools for accommodation of uncertainties nonlinearities. And can avoid wasteful cancellations. However, the not adaptive backstepping approach is capable of keeping almost all the robustness properties of the mismatched uncertainties. The not adaptive backstepping is a rigorous and procedure design methodology for nonlinear feedback control. The principal idea of this approach is to recursively design controllers for machine torque constant uncertainty subsystems in the structure and ‘‘step back’’ the feedback signals towards the control input. This approach is different from the approach of the conventional feedback linearization in that it can avoid cancellation of useful nonlinearities in pursuing the objectives of stabilization and tracking. 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.

The speed variation of the PMSM is widely used in high-performance applications. The PMSM has very large power density, high power factor and high efficiency. In a high-performance control of PMSM, the information of rotor position and speed is very important. In the speed control loop, for the field oriented control, the coordinate transformation has needs precise rotor position. Rotor position and speed can be measured by a shaft encoder or other type of sensors, in other case the speed is measured with an Encoder resolver connected to the PMSM machine drive. However, the presence of such sensors is not acceptable for cost, maintenance and reliability reasons. The concept of sensorless control was proposed in the 1970s and has been continually developed for PMSM rotor position and speed estimation. The basic principle of sensorless control is to deduce the rotor speed and position using various information and means, including direct calculation, parameter identification, condition estimation, indirect measuring and so on. The stator currents and voltages are generally used to calculate the information of speed and rotor position.

The FPGA technology is now used by an increasing number of designers in various fields of application such as signal processing, 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 [1]. 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.

Considering the complexity of the diversity of the electric control devices of the machines, it is difficult to define with universal manner a general structure for such systems.

Considering the complexity of the diversity of the electric control devices of the machines, it is difficult to define with universal manner a general structure for such systems. However, by having a reflexion compared to the elements most commonly encountered in these systems, it is possible to define a general structure of an electric control device of machines which is show in *Fig.1*:

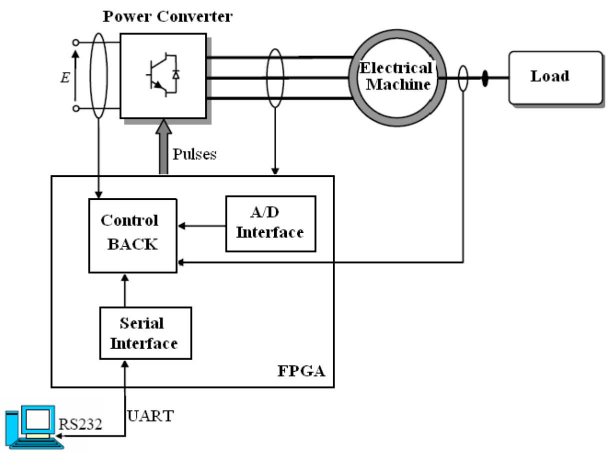


Fig.1: Architecture of the Control

This paper presents the realization of a platform for not adaptive and adaptive 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, Encoder interface and RS232 UART architecture [2].

2. PMSM model system

In this paper, we apply the different algorithms control on a machine type PMSM (Permanent Magnet Synchronous Motor) [3], which consists of three stator windings and a rotor magnet. This motor is described by the following equation (Voltage, Flux, Torque…),

$$\begin{cases} V_{sd} = r_s.i_{sd} + \dfrac{d\Phi_{sd}}{dt} - \omega.\Phi_{sq} \\ V_{sq} = r_s.i_{sq} + \dfrac{d\Phi_{sq}}{dt} - \omega.\Phi_{sd} \\ \Phi_{sd} = L_{sd}.i_{sd} + \Phi_f \\ \Phi_{sq} = L_{sq}.i_{sq} \\ C_e = J.\dfrac{d\Omega}{dt} + f.\Omega - C_r \\ \omega = p.\Omega \end{cases} \tag{1}$$

Where Ω is the rotation's speed, p the Number of pairs of poles, J the moment of inertia, f the Coefficient of viscous friction, C_r the resistive torque, Φ_f the flux produced by the permanent magnet, L_{sd} and L_{sq} the d-q axis stator inductance, V_{sd} and V_{sq} the d-q axis stator voltage, r_s the stator winding resistance and C_e the electromagnetic torque.

3. Nonlinear not adaptive Backstepping approach

The schematic diagram of the speed control system study with application of the nonlinear Backstepping controller is shown in Fig.1. The parameters of the synchronous machine are given in the Appendix. In this section, we employ the nonlinear Backstepping schemes to design the controllers for PMSM systems with angular velocity measurement.

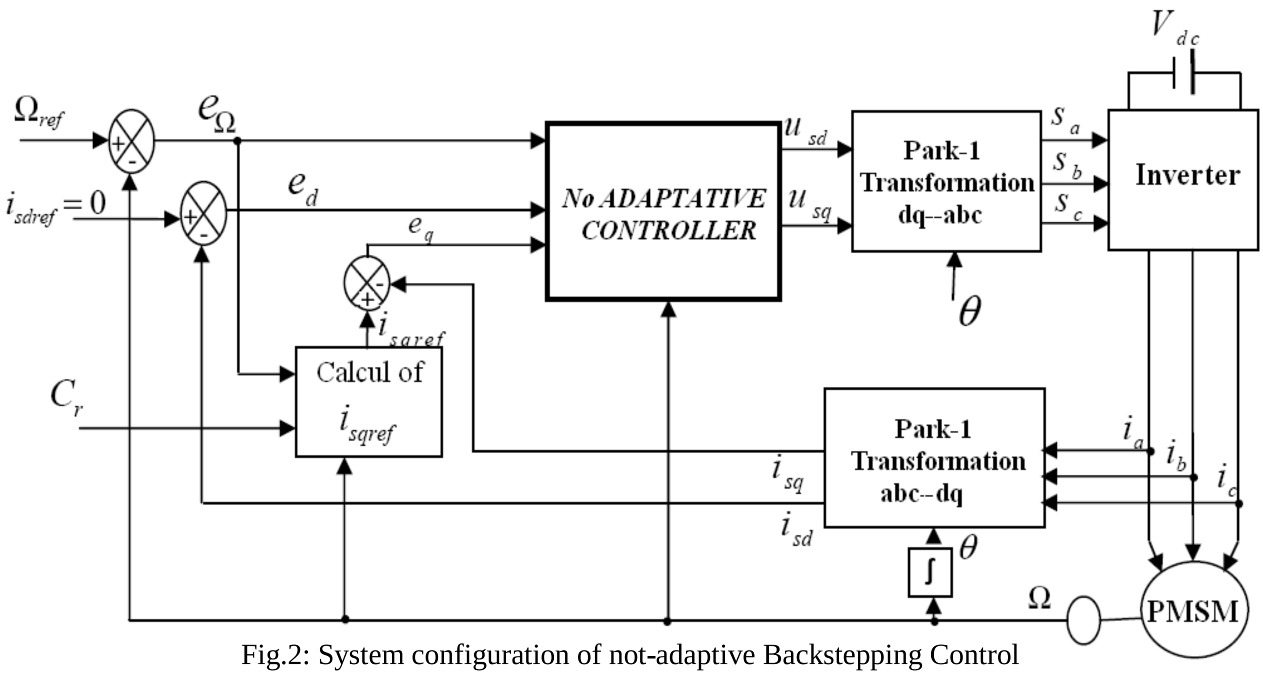


Fig.2: System configuration of not-adaptive Backstepping Control

The Not Adaptive Backstepping approach algorithm is control techniques that can linearizing a nonlinear system such as the PMSM machine drive in the presence of uncertainties. Unlike other feedback linearization techniques, adaptive Backstepping has the flexibility of keeping useful non linearity's 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 [4].

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

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

The vector $[x]=[i_{sd} \quad i_{sq} \quad \Omega]^T$ 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 principal objective of the backstepping controller is to regulate the speed of the PMSM drive to its reference value Ω_{ref} whatever external disturbances. We assume that the engine parameters are known and invariant.

3.1. Backstepping Speed Controller

The first step is defined the tracking errors:

$$e_\Omega = \Omega_{ref} - \Omega \tag{3}$$

The derivative of (3) is:

$$\begin{aligned}
\dot{e}_\Omega &= \frac{de_\Omega}{dt} = \dot{\Omega}_{ref} - \dot{\Omega} \\
&= \dot{\Omega}_{ref} - \frac{1}{J}\left[\frac{3p}{2}(\Phi_f i_{sq} + (L_{sd}-L_{sq})i_{sd}i_{sq}) - f\Omega - C_r\right]
\end{aligned} \tag{4}$$

We define the following quadratic function:

$$V_1 = \frac{1}{2}e_\Omega^2 \tag{5}$$

Its derivative along the solution of (5), is given by:

$$\begin{aligned}
\dot{V}_1 &= e_\Omega \dot{e}_\Omega \\
&= e_\Omega\left(\dot{\Omega}_{ref} - \frac{1}{J}\left[\frac{3p}{2}(\Phi_f i_{sq} + (L_{sd}-L_{sq})i_{sd}i_{sq}) - f\Omega - C_r\right]\right)
\end{aligned} \tag{6}$$

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:

$$\begin{cases} i_{sdref} = 0 \\ i_{sqref} = \dfrac{2}{3p\Phi_f}(f\Omega + C_r + J.k_\Omega.e_\Omega) \end{cases} \tag{7}$$

With k_Ω is a positive constant

Substituting (7) in (6) the derivative of V_1:

$$\dot{V}_1 = -k_\Omega e_\Omega^2 \le 0 \tag{8}$$

3.2. Backstepping Current Controller

We have the asymptotic stability of the origin of the system (1). We defined current following errors:

$$\begin{cases} e_d = i_{sdref} - i_{sd} \quad \text{with} \quad i_{sdref} = 0 \\ e_q = i_{sqref} - i_{sq} \end{cases} \tag{9}$$

Their dynamics can be written:

$$\dot{e}_d = \dot{i}_{sdref} - \dot{i}_{sd} = \frac{r_s}{L_{sd}}.i_{sd} - \frac{L_{sq}}{L_{sd}}p\Omega i_{sq} - \frac{V_{sd}}{L_{sd}}\dot{e}_q$$

$$= \dot{i}_{sqref} - \dot{i}_{sq} = \frac{2}{3p\Phi_f}(f\dot{\Omega} + \dot{C}_r + J.k_\Omega.\dot{e}_\Omega) \tag{10}$$

$$+\frac{r_s}{L_{sq}}.i_{sq} + \frac{L_{sd}}{L_{sq}}p\Omega i_{sd} + \frac{p\Phi_f}{L_{sq}}\Omega - \frac{V_{sd}}{L_{sd}}$$

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

$$V_2 = \frac{1}{2}(e_\Omega^2 + e_d^2 + e_q^2) \tag{11}$$

Its derivative along the trajectories (8), (9) and (10) is:

$$\dot{V}_2 = e_\Omega\dot{e}_\Omega + e_d\dot{e}_d + e_q\dot{e}_q = -k_\Omega e_\Omega^2 - k_d e_d^2 - k_q e_q^2 +$$

$$e_d[k_d e_d - \frac{V_{sd}}{L_{sd}} + \frac{r_s}{L_{sd}} - \frac{L_{sq}}{L_{sd}}\Omega i_{sq} + \frac{3p}{2J}(L_{sd} - L_{sq})e_\Omega i_{sq}]$$

$$+e_q[k_q e_q + \frac{2(k_\Omega J - f)}{3p\Phi_f}.(\frac{3p\Phi_f}{2J}e_q + \frac{3p}{2J}(L_{sd} - L_{sq})e_d i_{sq} - k_\Omega e_\Omega) \tag{12}$$

$$+\frac{3p\Phi_f}{2J}e_\Omega - \frac{V_{sq}}{L_{sq}} + \frac{r_s}{L_{sq}}i_{sq} + \frac{L_{sd}}{L_{sq}}\Omega i_{sd} + \Omega\frac{\Phi_f}{L_{sq}}]$$

The expression (12) found above requires the following control laws:

$$V_{sd} = k_d L_{sd} e_d + r_s i_{sd} - L_{sq}\Omega i_{sq} + \frac{3pL_{sd}}{2J}(L_{sd} - L_{sq})e_\Omega i_{sq}$$

$$V_{sq} = \frac{2L_{sq}(k_\Omega J - f)}{3p\Phi_f}\left(\frac{3p\Phi_f}{2J}e_q + \frac{3p}{2J}(L_{sd} - L_{sq})e_d i_{sq} - k_\Omega e_\Omega\right) \tag{13}$$

$$+\frac{3p\Phi_f L_{sq}}{2J}e_\Omega + r_s i_{sq} + L_{sd}\Omega i_{sd} + \Omega\Phi_f + k_q L_{sq} e_q$$

With this choice the derivatives of (13) become:

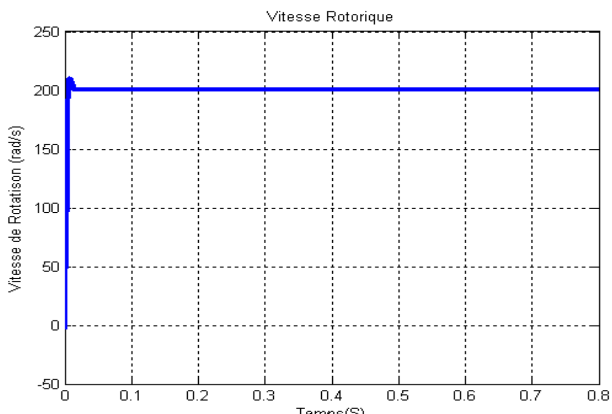
$$\dot{V}_2 = -k_\Omega e_\Omega - k_d e_d - k_q e_q \le 0 \tag{14}$$

3.3. Simulation and test performance

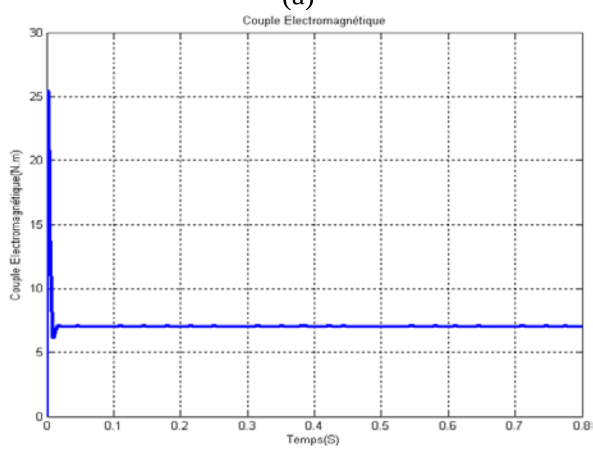
In this section, we show the great improvement of performance PMSM machine, at first the results of not adaptive Backstepping control is analyzed. After, the performance of the indirect sliding mode control is discussed and compared to that before.The characteristics for the PMSM motor simulated in this experiment are follows:

r_s=0.412 *Ω, p=4,* L_{Sd}*=3,24 mH,* L_{Sq}*=3,28 mH, J=0,0001473 Kg,m*^{2,}

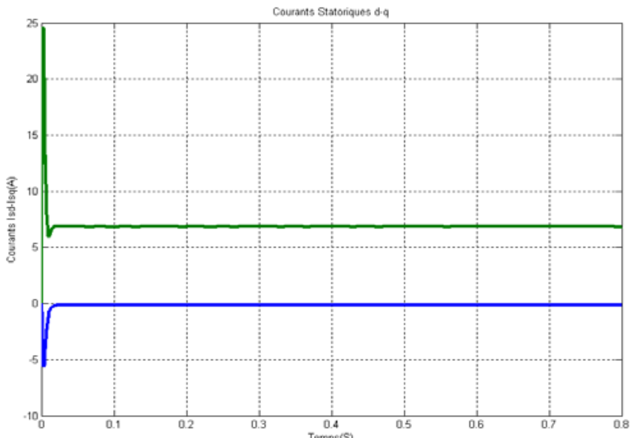
For a trajectory Ω_{ref}=150 rad/s at 0s, i_{sdref}=4.5A and C_r=7N.m.the following (*Fig.3*) shown the performance of the input output linearization control.



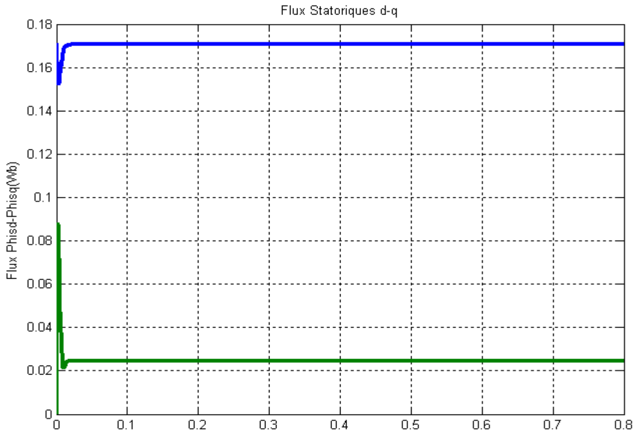
(a)



(b)



(c)

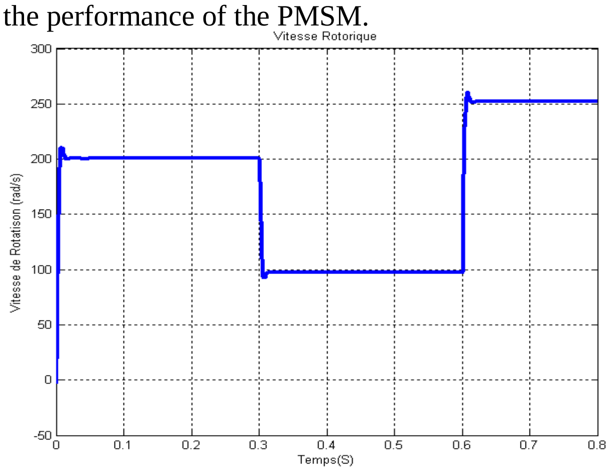


(d)

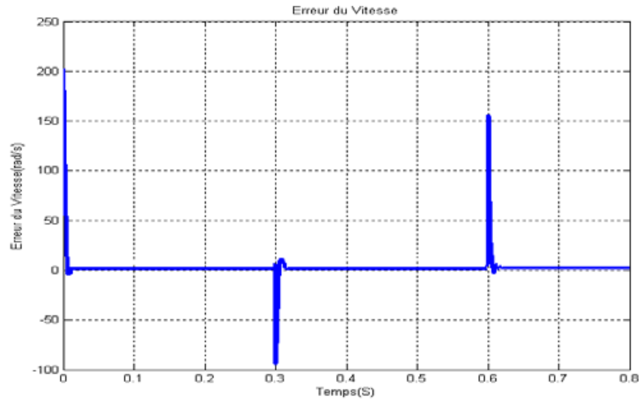
Fig.3: (a) Speed response trajectory, (b) Electromagnetic Torque C_e, (c) d-q axis current without uncertainties, (d) d-q axis Flux without uncertainties

- **Follow of the trajectory**

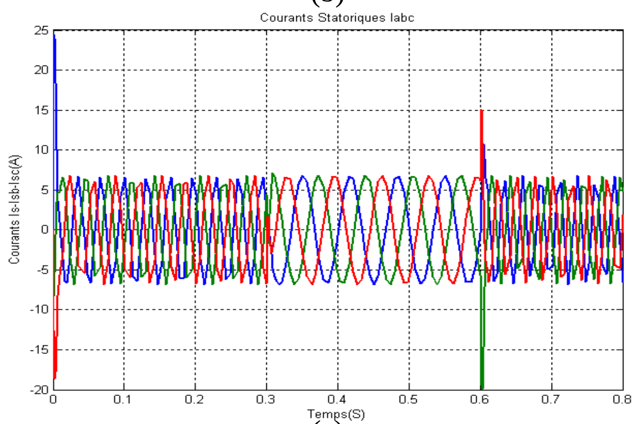
For test the further trajectory, the speed reference was made variable. The reference of the direct component of current was set to zero. For a trajectory Ω_{ref} (=200rad/s at 0s, =100rad/s at t=0.3s, =250rad/s at t=0.6s), the following (*Fig.4*) shown the performance of the PMSM.



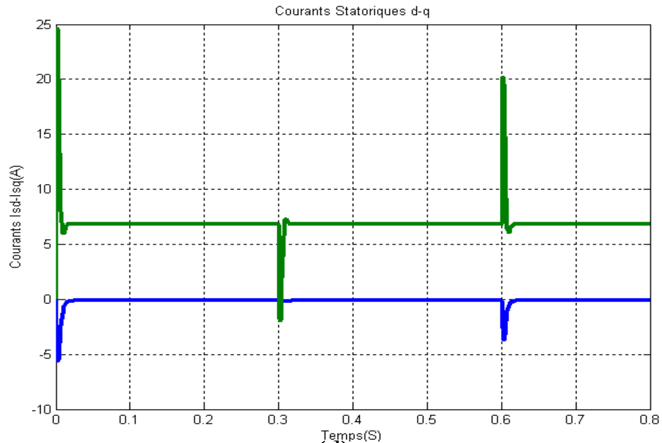
(a)



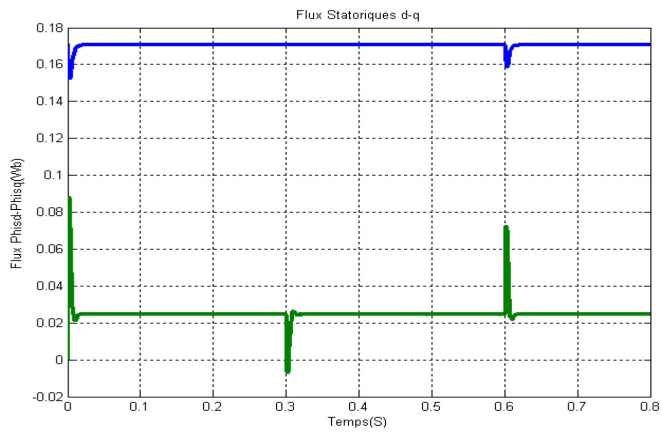
(b)



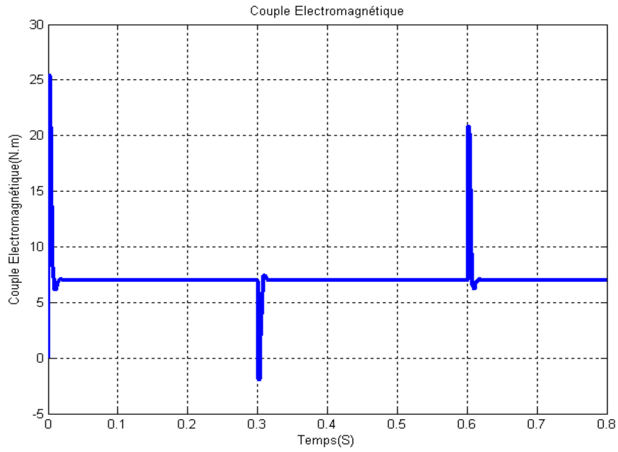
(c)



(d)



(e)

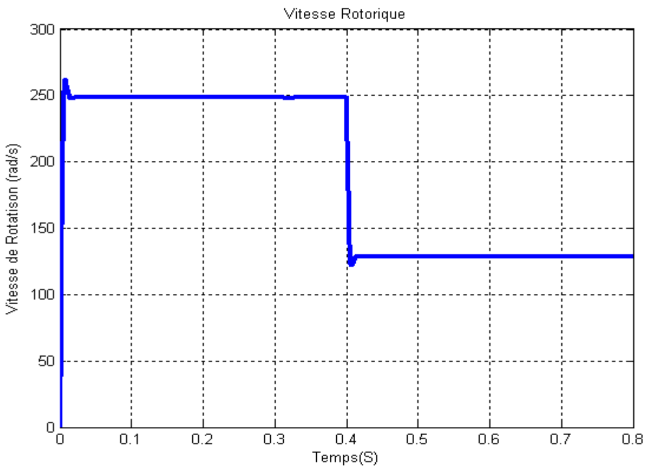


(f)

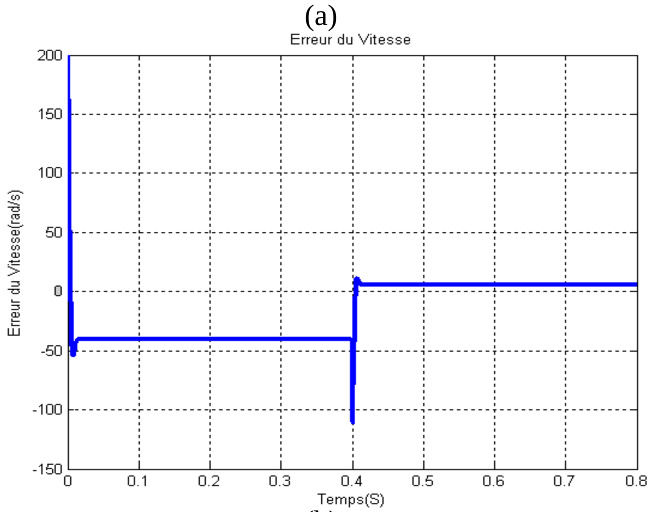
Fig.3: (a) Speed response trajectory, (b) Error Speed response, (c) abc Axis Stator Current, (d) dq Axis Stator Current, (e) d-q Axis Stator Flux, (f) Electromagnetic Torque C_e

➢ Disturbance rejection torque load

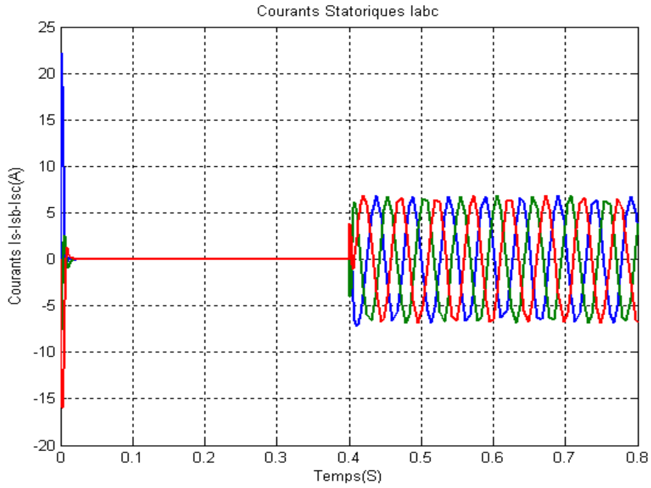
For a trajectory Ω_{ref} (=250rad/s at 0s, =130rad/s at t=0.4s) and Cr=7 N.m at t=0.4s, the following (*Fig.5*) shown the performance of the PMSM.



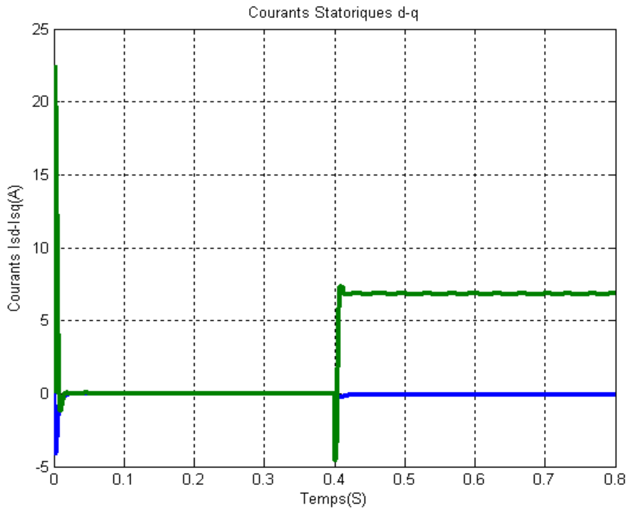
(a)



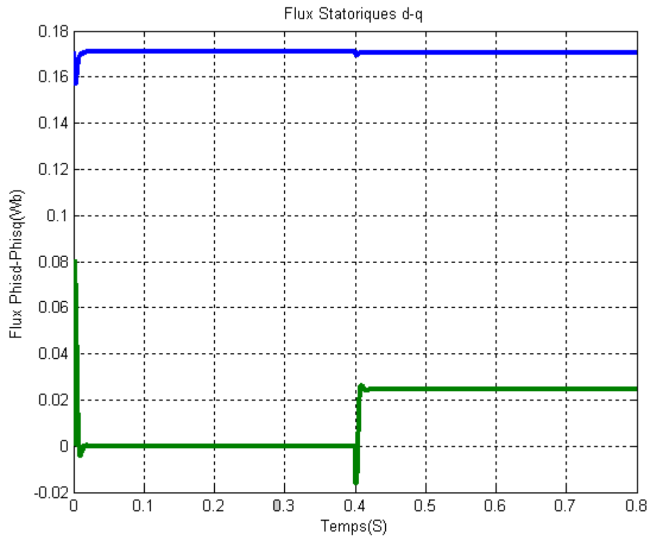
(b)



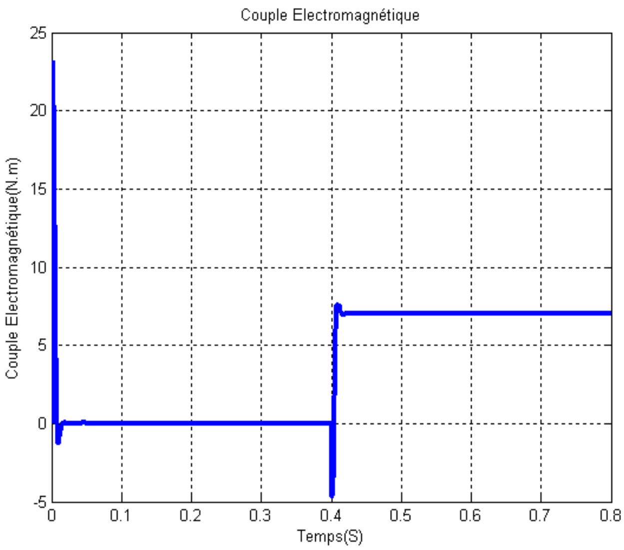
(c)



(d)



(e)



(f)

Fig.5: (a) Speed response trajectory, (b) Error Speed response, (c) abc Axis Stator Current, (d) dq Axis Stator Current, (e) d-q Axis Stator Flux, (f) Electromagnetic Torque C_e

4. Nonlinear adaptive Backstepping approach Control

4.1. Principle

In the previous section, the control laws are developed under the assumption that the machine parameters are known and invariants. This assumption is not always true. In fact, the flow created by the magnet varies with increasing temperature and with the fields created by the stator currents. Stator resistance also varies with temperature. Also, the change in operating conditions is implicitly load torque and hence the coefficient of friction and inertia. Adaptive Backstepping version takes into account the variations of these parameters.

In equation (7), we do not know exactly the value of the load torque C_r, it will be replaced by its estimate $\hat{C}_r$.

$$\hat{i}_{sqref} = \frac{2}{3p\Phi_f}(f.\Omega + \hat{C}_r + J.k_\Omega.e_\Omega) \quad (15)$$

From (13) and (15), we deduce the dynamics of the speed error as follows:

$$\frac{de_\Omega}{dt} = \frac{1}{J}\begin{pmatrix} \tilde{C}_r + \dfrac{3p\Phi_f}{2}e_q + \\ \dfrac{3p}{2}(L_{sd} - L_{sq})e_d i_{sq} - J.k_\Omega.e_\Omega \end{pmatrix} \quad (16)$$

With $\tilde{C}_r = \hat{C}_r - C_r$ is the error of the estimated load torque.

The Dynamic errors and direct currents quadratic write:

$$\frac{de_\Omega}{dt} = -\frac{di_{sd}}{dt} = -\frac{V_{sd}}{L_{sd}} + \frac{R_s}{L_{sd}}i_{sd} - \Omega\frac{L_{sq}}{L_{sd}}i_{sq} \quad (17)$$

$$\frac{de_q}{dt} = \frac{di_{sqref}}{dt} - \frac{di_{sq}}{dt}$$

$$= \frac{2(k_\Omega J - f)}{3p\Phi_f}\left[\frac{3p\Phi_f}{2J}e_q + \frac{3p}{2J}(L_{sd} - L_{sq})e_d i_{sq} - k_\Omega.e_\Omega\right] - \quad (18)$$

$$-\frac{V_{sd}}{L_{sq}} + \frac{R_s}{L_{sq}}i_{sd} + \Omega\frac{L_{sd}}{L_{sq}}i_{sd} + \Omega\frac{\Phi_f}{L_{sq}} + \frac{2}{3p\Phi_f}(\frac{f}{J} - k_\Omega)\tilde{C}_r$$

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

$$V_2 = \frac{1}{2}\left(e_\Omega^2 + e_d^2 + e_q^2 + \frac{\tilde{C}_r^2}{\gamma_1} + \frac{\tilde{R}_s^2}{\gamma_2} + \frac{\tilde{\Phi}_f^2}{\gamma_3}\right) \quad (19)$$

Its derivative along the trajectories (16), (17) and (18) is:

$$\dot{V}_2 = e_\Omega\dot{e}_\Omega + e_d\dot{e}_d + e_q\dot{e}_q + \frac{1}{\gamma_1}\tilde{C}_r\dot{\tilde{C}}_r + \frac{1}{\gamma_2}\tilde{R}_s\dot{\tilde{R}}_s + \frac{1}{\gamma_3}\tilde{\Phi}_f\dot{\tilde{\Phi}}_f$$

$$= -k_\Omega e_\Omega^2 - k_d e_d^2 - k_q e_q^2$$

$$+ e_d\left[k_d e_d - \frac{V_{sd}}{L_{sd}} + \frac{R_s}{L_{sd}} - \frac{L_{sq}}{L_{sd}}\Omega i_{sq} + \frac{3p}{2J}(L_{sd} - L_{sq})e_\Omega i_{sq}\right]$$

$$+ e_q[k_q e_q + \frac{2(k_\Omega J - f)}{3p\hat{\Phi}_f}.(\frac{3p\hat{\Phi}_f}{2J}e_q + \frac{3p}{2J}(L_{sd} - L_{sq})e_d i_{sq}$$

$$-k_\Omega e_\Omega) + \frac{3p\hat{\Phi}_f}{2J}e_\Omega - \frac{V_{sq}}{L_{sq}} + \frac{R_s}{L_{sq}}i_{sq} + \frac{L_{sd}}{L_{sq}}\Omega i_{sd} + \Omega\frac{\hat{\Phi}_f}{L_{sq}}]$$

$$+ \tilde{C}_r\left[\frac{1}{\gamma_1}\dot{\tilde{C}}_r - \frac{2k_\Omega e_q}{3p\hat{\Phi}_f} + \frac{2fe_q}{3pJ\hat{\Phi}_f} - \frac{e_\Omega}{J}\right]$$

$$+ \tilde{R}_s\left[\frac{1}{\gamma_2}\dot{\tilde{R}}_s + \frac{1}{L_{sd}}e_d i_{sd} - \frac{1}{L_{sq}}e_q i_{sq}\right]$$

$$+ \tilde{\Phi}_f\left[\frac{1}{\gamma_3}\dot{\tilde{\Phi}}_f - \frac{3p}{2J}e_\Omega e_q - \frac{k_\Omega J - f}{J\hat{\Phi}_f}e_q^2 - \frac{1}{L_{sq}}\Omega e_q\right] \quad (20)$$

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

$$V_{sd} = k_d L_{sd}e_d + \hat{R}_s i_{sd} - L_{sq}\Omega i_{sq} + \frac{3pL_{sd}}{2J}(L_{sd} - L_{sq})e_\Omega i_{sq}$$

$$V_{sq} = \frac{2L_{sq}(k_\Omega J - f)}{3p\Phi_f}\left(\frac{3p\hat{\Phi}_f}{2J}e_q + \frac{3p}{2J}(L_{sd} - L_{sq})e_d i_{sq} - k_\Omega e_\Omega\right) \quad (21)$$

$$+\frac{3p\hat{\Phi}_f L_{sq}}{2J}e_\Omega + \hat{R}_s i_{sq} + L_{sd}\Omega i_{sd} + \Omega\hat{\Phi}_f + k_q L_{sq}e_q$$

Therefore the dynamics of the Lyapunov function can be simplified as follows:

$$\begin{aligned}\dot{V}_2 &= -k_\Omega e_\Omega^2 - k_d e_d^2 - k_q e_q^2 \\ &+ \tilde{C}_r\left[\frac{1}{\gamma_1}\dot{\tilde{C}}_r - \frac{2k_\Omega e_q}{3p\hat{\Phi}_f} + \frac{2fe_q}{3pJ\hat{\Phi}_f} - \frac{e_\Omega}{J}\right] \\ &+ \tilde{R}_s\left[\frac{1}{\gamma_2}\dot{\tilde{R}}_s + \frac{1}{L_{sd}}e_d i_{sd} - \frac{1}{L_{sq}}e_q i_{sq}\right] \\ &+ \tilde{\Phi}_f\left[\frac{1}{\gamma_3}\dot{\tilde{\Phi}}_f - \frac{3p}{2J}e_\Omega e_q - \frac{k_\Omega J - f}{J\hat{\Phi}_f}e_q^2 - \frac{1}{L_{sq}}\Omega e_q\right]\end{aligned} \tag{22}$$

Hence the adaptation laws as follows:

$$\dot{\tilde{C}}_r = \gamma_1\left[\frac{2k_\Omega e_q}{3p\hat{\Phi}_f} - \frac{2fe_q}{3pJ\hat{\Phi}_f} + \frac{e_\Omega}{J}\right] \tag{23}$$

$$\dot{\tilde{R}}_s = \gamma_2\left[-\frac{1}{L_{sd}}e_d i_{sd} + \frac{1}{L_{sq}}e_q i_{sq}\right] \tag{24}$$

$$\dot{\tilde{\Phi}}_f = \gamma_3\left[\frac{3p}{2J}e_\Omega e_q + \frac{k_\Omega J - f}{J\hat{\Phi}_f}e_q^2 + \frac{1}{L_{sq}}\Omega e_q\right] \tag{25}$$

With this choice, the expression (19) becomes:

$$\dot{V}_2 = -k_\Omega e_\Omega^2 - k_d e_d^2 - k_q e_q^2 \le 0 \tag{26}$$

So the system is globally asymptotically stable in the presence of parametric uncertainties.

4.2. Simulation and test performance

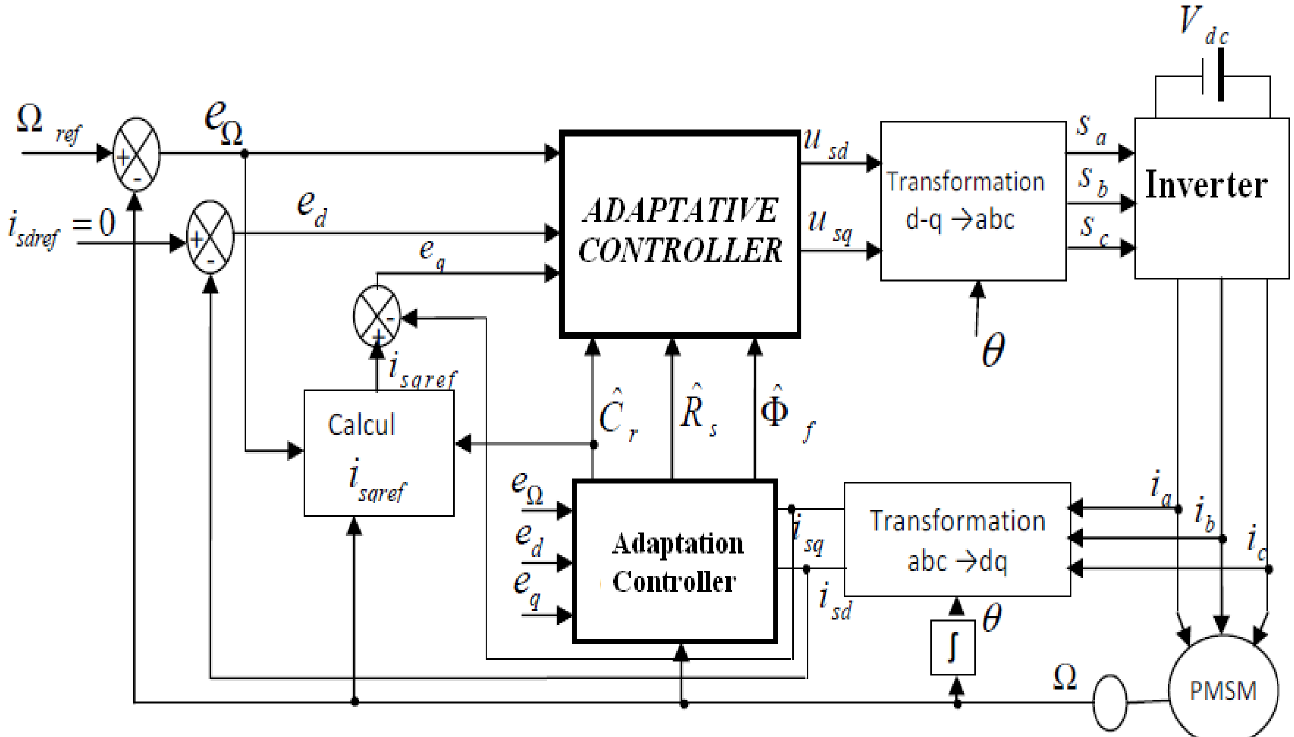
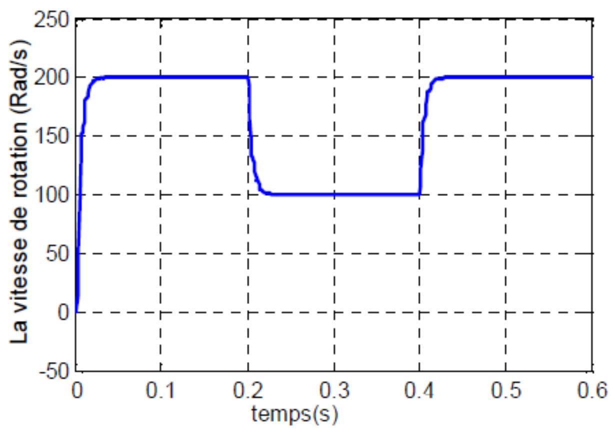


Fig.6: System configuration of adaptive Backstepping Control

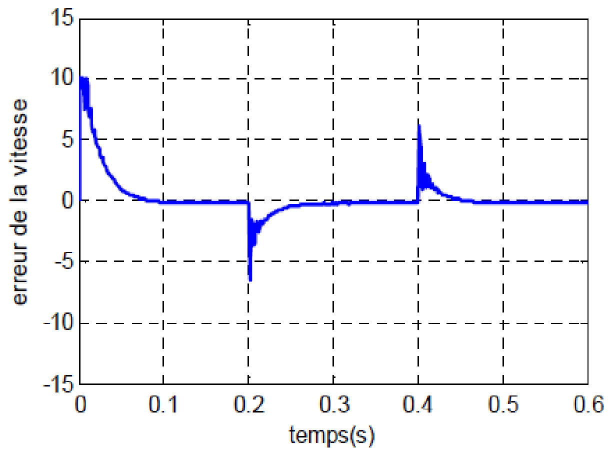
The following results are obtained by choosing the following values:

- ✓ Gains of the control law: $k_\Omega = 0.15$, $k_d = 0.01$, $k_q = 0.01$.
- ✓ Adaptation gains: $\gamma_1 = 0.15$, $\gamma_2 = 0.01$, $\gamma_3 = 0.015$.

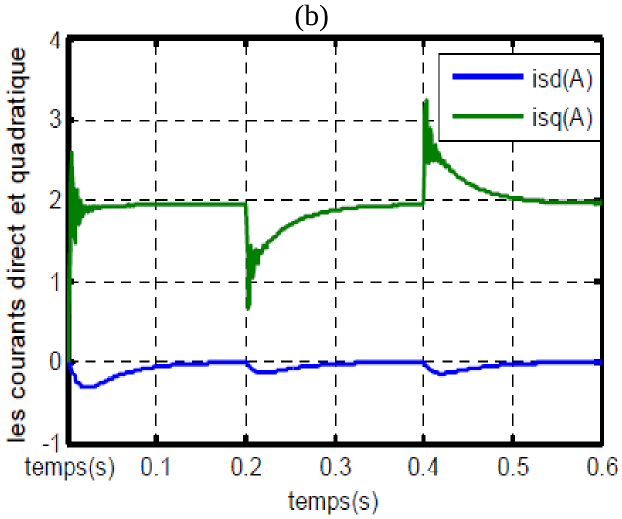
➢ **Follow of the trajectory**



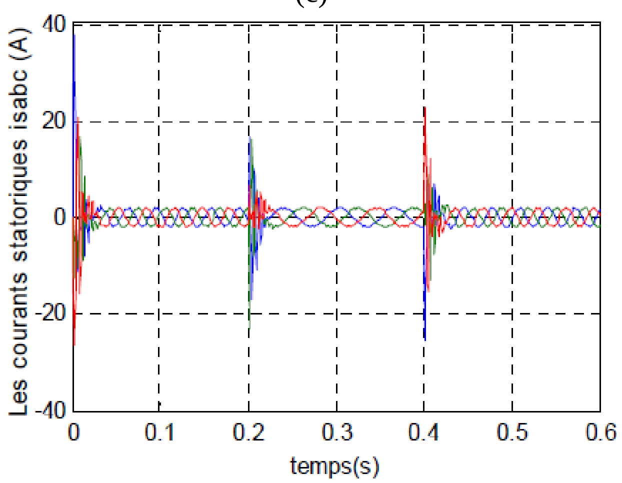
(a)



(b)



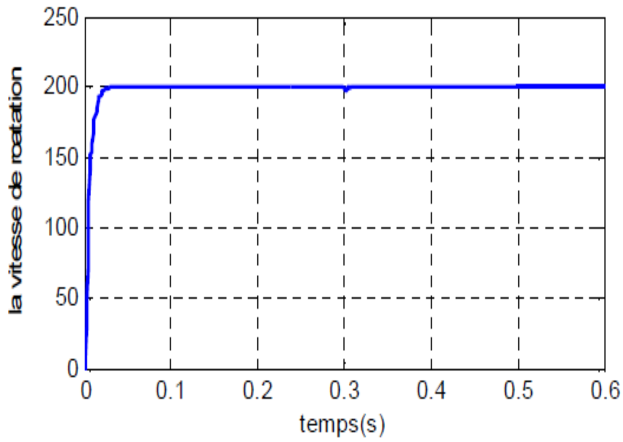
(c)



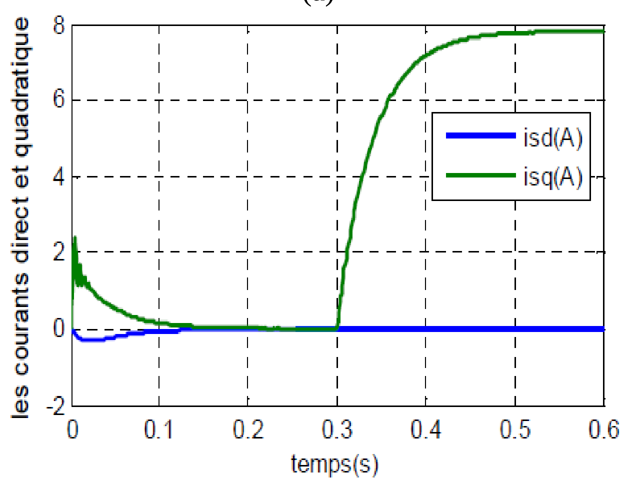
(d)

Fig.7: Test performance of the adaptive controller for trajectory tracking
(a) Speed response trajectory (b) Error Speed response (c) d-q axis current without uncertainties (d) abc axis current

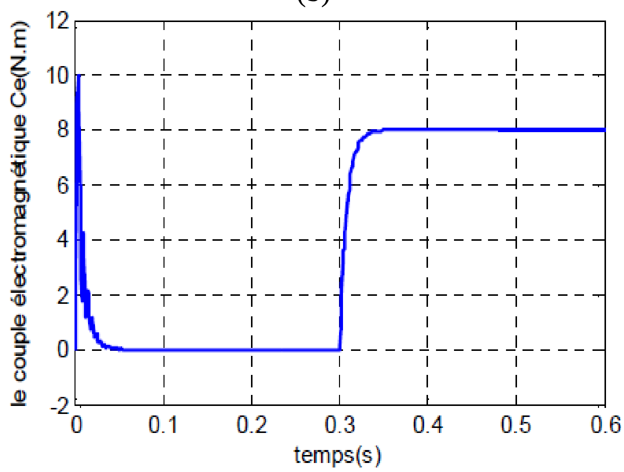
➢ **Disturbance rejection**



(a)



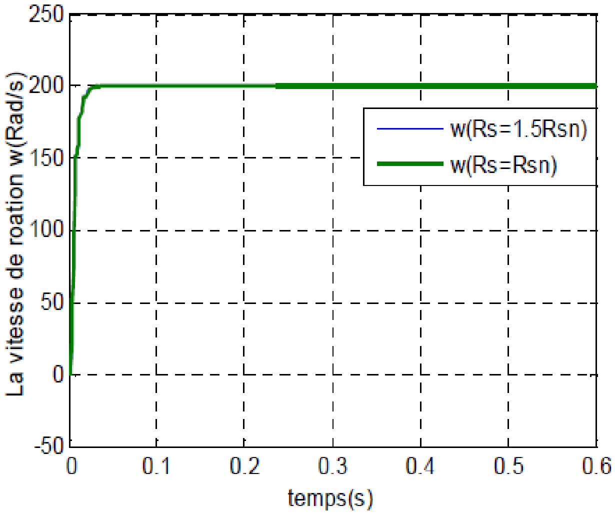
(b)



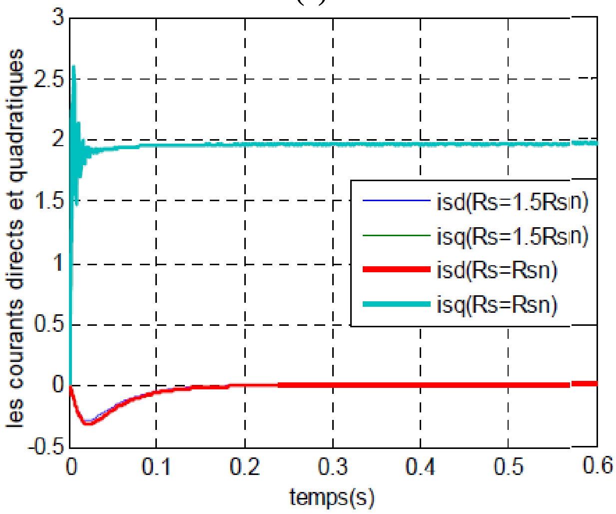
(c)

Fig.8: Test performance of the adaptive controller for rejecting disturbance torque load applied at t = 0.3s.
(a) Speed response trajectory (b) d-q axis current without uncertainties (c) Electromagnetic Torque

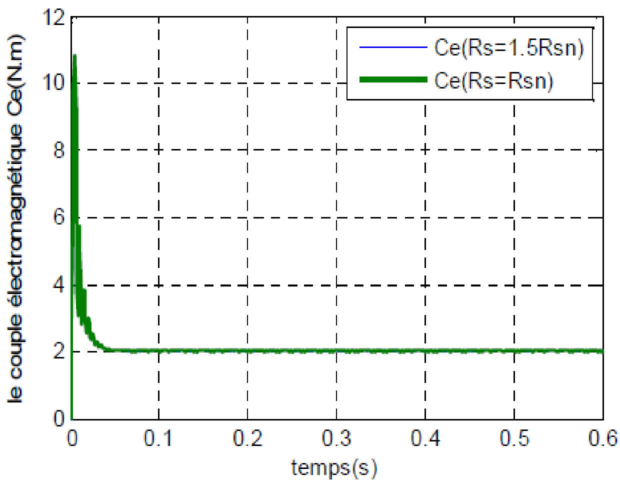
- **Parametric uncertainties**



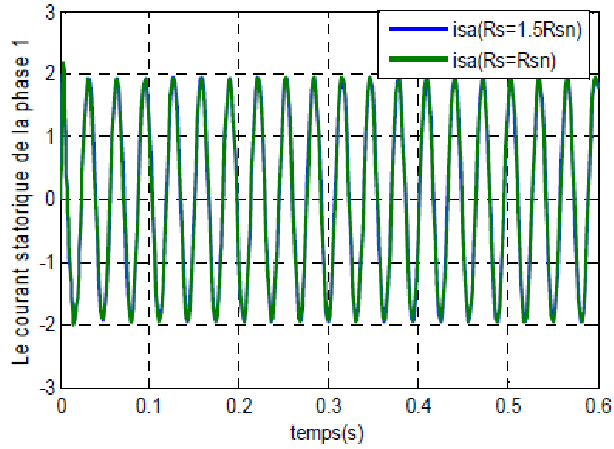
(a)



(b)

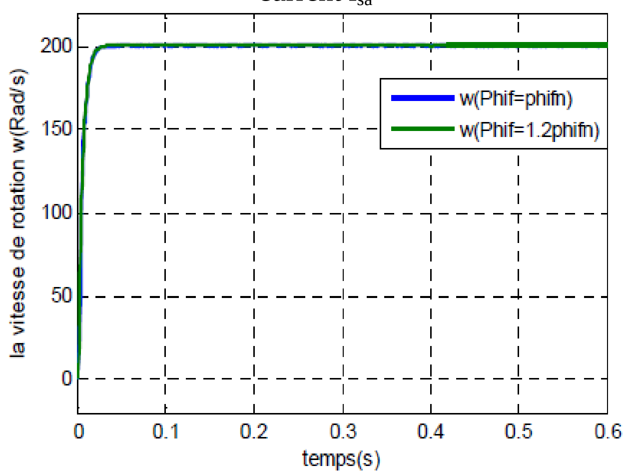


(c)

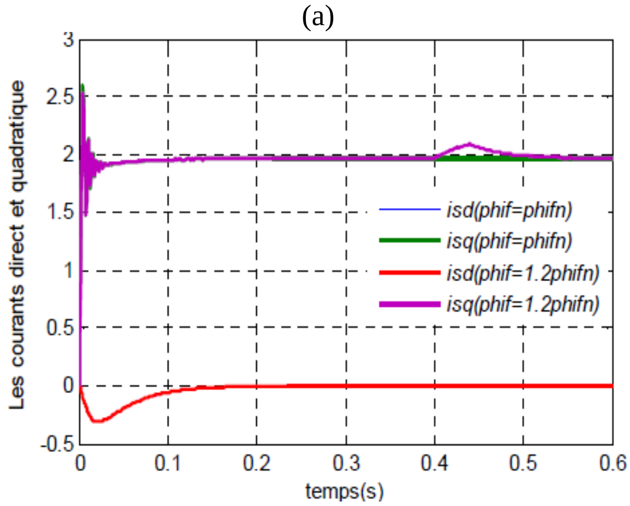


(d)

Fig.9: Test performance of the adaptive controller following a change in R_s
(a) Speed response trajectory (b) d-q axis current without uncertainties (c) Electromagnetic Torque (d) current i_{sa}



(a)



(b)

Fig.10: Test performance of the adaptive controller following a change in Φ_f
(a) Speed response trajectory (b) d-q axis current without uncertainties

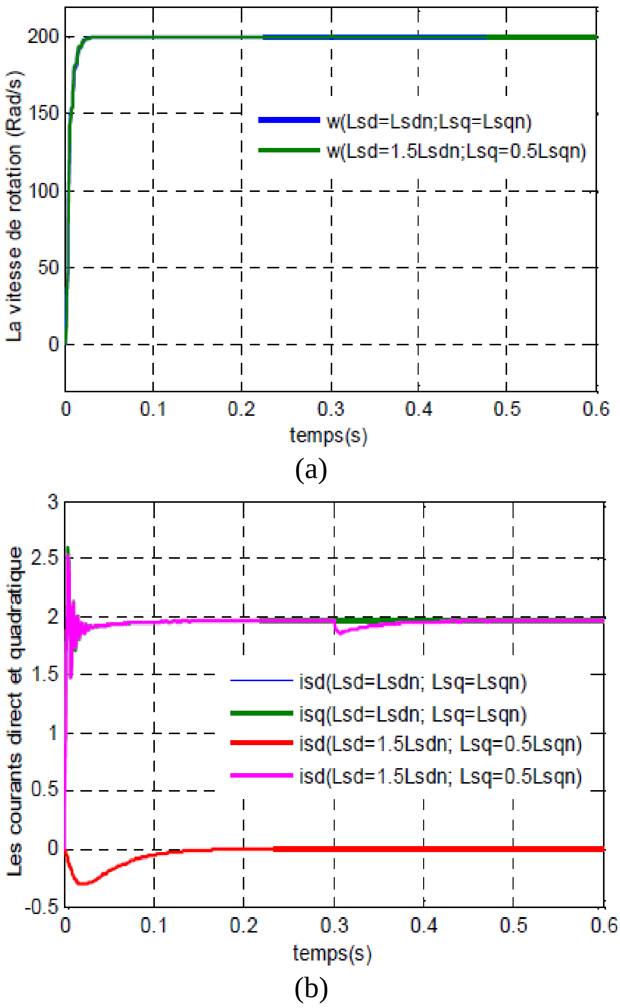


Fig.11: Test performance of the adaptive controller following a change in L_{sd} and L_{sq}
Speed response trajectory (b) d-q axis current without uncertainties

5. FPGA-Based Implementation of an robust Backstepping control system

5.1. 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 [7].

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.

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 adaptive Backstepping* algorithm associated with a voltage inverter and PMSM model.

The *Fig.12* 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.

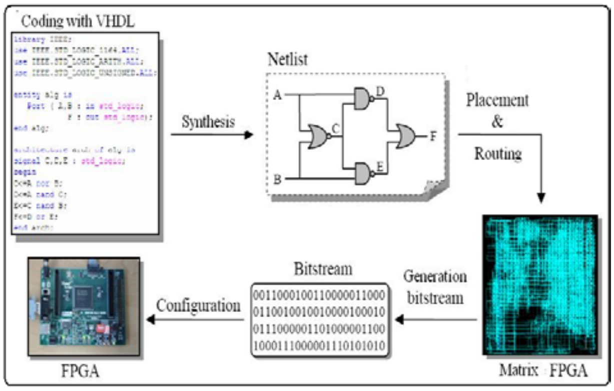


Fig.12: Programming FPGA devisees

In this work 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.

5.2. 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 time discretired model of the No adaptive Backstepping algorithm associated with a voltage inverter and PMSM model. The *Fig.12* shows in detail the programming of the control shown in Fig.2 in the SYSTEM GENERATOR environment from Xilinx, we will implement it later in the memory of the FPGA for the simulation of PMSM.

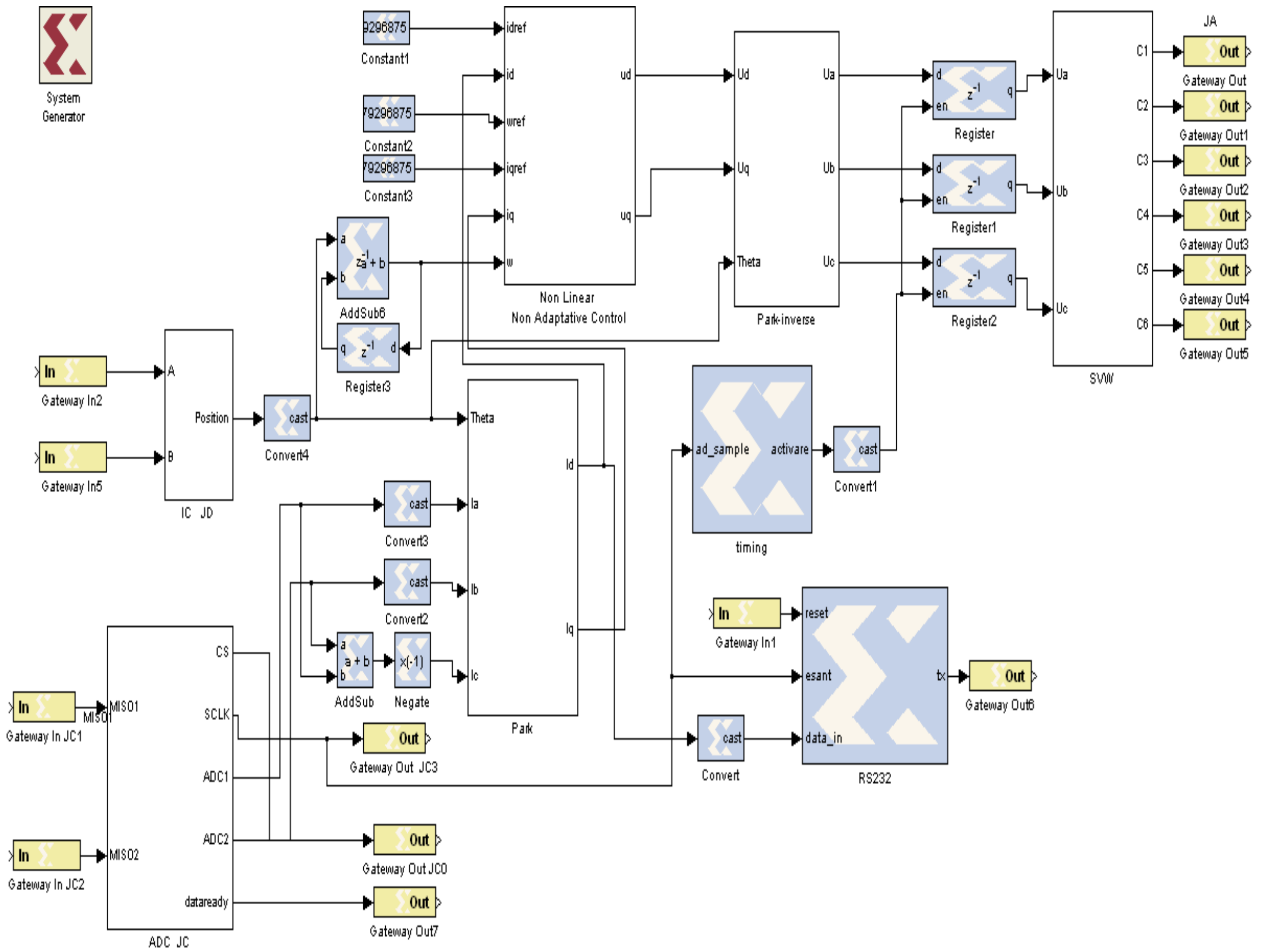


Fig.13: Functional Model for No adaptive Backstepping Controller from SYSTEM GENERATOR

The functional model for no adaptive 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 adaptive 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 fixed point format. The *Fig.14* gives the specification model of the *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):

$$\begin{bmatrix} V_{sd} \\ V_{sq} \\ V_0 \end{bmatrix} = \sqrt{\frac{2}{3}} \begin{bmatrix} Cos\theta & Cos(\theta - \frac{2\pi}{3}) & Cos(\theta - \frac{4\pi}{3}) \\ -Sin\theta & -Sin(\theta - \frac{4\pi}{3}) & -Sin(\theta - \frac{4\pi}{3}) \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} V_{sa} \\ V_{sb} \\ V_{qc} \end{bmatrix} \quad (26)$$

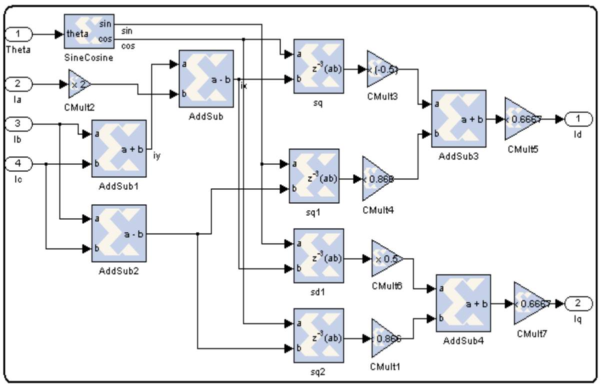
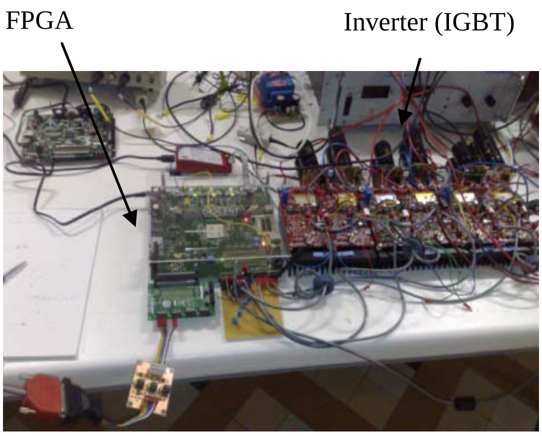


Fig.14: Park Transformation Model.

The specification model is then used for the definition of the corresponding Data Flow Graph. The *Fig.14* shows the *DFG* corresponding to the Park transformation module.

5.3. Prototyping platform

To test the *FPGA* based controller, a prototyping platform for the control of a Permanent magnet Synchronous Machine was assembled (*Fig.15*).



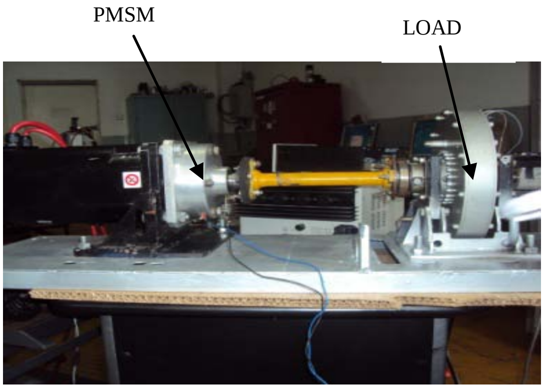
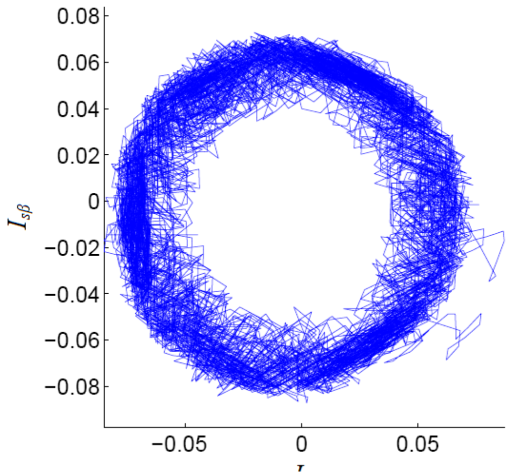


Fig.15: Prototyping platform control

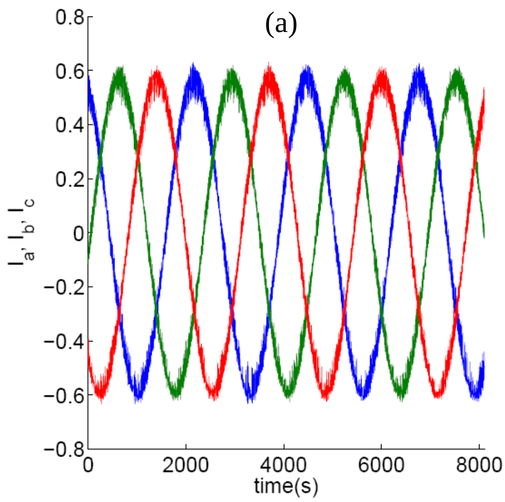
6. Experimental results

The implementation of the indirect control by sliding mode on *FPGA* devices is characterized by a reduced operation time.

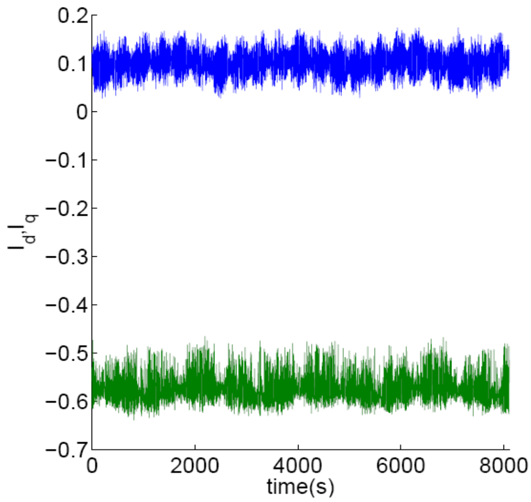
The *Fig.16* shown the experimental results of Indirect Sliding Mode PMSM with the FPGA platform are shown. Update frequency for this implementation is 20 kHz. All results were extracted from the FPGA by the ChipScope tool of Xilinx.



(a)



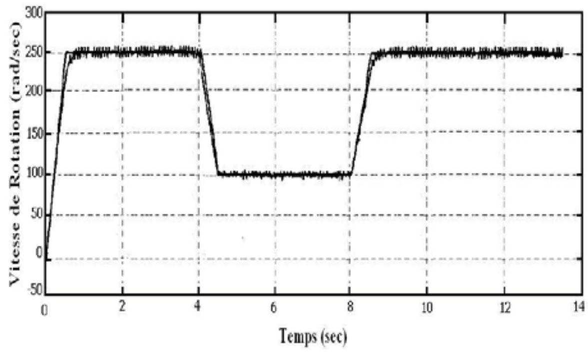
(b)



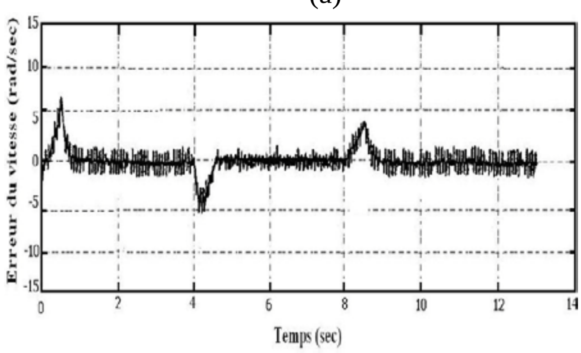
(c)

Fig.16: (a) Stator current locus for ISMC, (b) abc-axis current in the PMSM, (c) d-axis and q-axis current in the PMSM

In *Fig.17.a* the experimental results No Adaptive 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.17.b* shows the stator current i_{sa} and i_{sb}. Update frequency for this implementation is 20 kHz.



(a)



(b)

Fig.16: (a) Speed performance rotor of PMSM, (b) Error Speed performance

The Experimental results show the performance of PMSM machine, using two approaches control nonlinear. This two control algorithms show the robustness and effecacité of the system.

7. Conclusion

In this paper a robust continuous approachs Nonlinear not Adaptive Backstepping Control and Adaptive Backstepping 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.

Acknowledgment

We thank all those who contributed to make this work, including my teachers in my home laboratory STIC Team and the *Center of Modeling and simulation of the systems*, my wife and all my friends for their support. This work was performed in the Montefiore Institute, University of Liege (ULG) in Belgium.

References

[1] Melicio, R; Mendes, VMF; Catalao, JPS "Modeling, Control and Simulation of Full-Power Converter Wind Turbines Equipped with Permanent Magnet SynchronousGenerator" vol 5, pg 397-408 Mar - Apr 2010.

[2] 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.

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

[4] 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.

[5] 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.

[6] 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.
[7] E. MONMASSON, and M. Cirstea “FPGA Design Methodology for Industrial Control Systems – A Review,” IEEE Trans Ind. Electron.., vol.54, no. 4, pp.1824-1842, August. 2007.
[8] B.BOSSOUFI, M.KARIM, S.IONITA, A.LAGRIOUI, “Low-Speed Sensorless Control of PMSM Motor drive Using a NONLINEAR Approach Backstepping Control: FPGA-Based Implementation” Journal of Theoretical and Applied Information Technology JATIT, pp154-166, Vol. 36 No.1, 29th February 2012.
[9] B.BOSSOUFI, M.KARIM, S.IONITA, A.LAGRIOUI, “Nonlinear Non Adaptive Backstepping with Sliding-Mode Torque Control Approach for PMSM Motor” Journal of Journal of Electrical Systems JES, pp236-248. Vol.8 No.2, June 2012.
[10] 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.
[11] 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.
[12] 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.
[13] 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.
[14] 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.
[15] 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.