Real time emulator for parallel connected dual-PMSM sensorless control

Khaldoune Sahri\textsuperscript{1}, Maria Pietrzak-David\textsuperscript{2}, Lothfi Baghli\textsuperscript{3}, Abdelaziz Kheloui\textsuperscript{4}

\textsuperscript{1-4}Laboratoire Commandes des Machines Electriques, Ecole Militaire Polytechnique, 16111 Bordj El Bahri, Algeria
\textsuperscript{2}Universit\'e de Toulouse; INPT, UPS, LAPLACE ENSEEIHT, CNRS, 2 rue Charles Camichel, F-31071 Toulouse cedex 7, France

Article Info

Article history:
Received Dec 9, 2019
Revised Mar 30, 2021
Accepted Jul 2, 2021

Keywords:
Emulator
FPGA
Multi-machine system
PMSM
Sensorless control

ABSTRACT

This paper presents a real-time emulator of a dual permanent magnet synchronous motor (PMSM) drive implemented on a field-programmable gate array (FPGA) board for supervision and observation purposes. In order to increase the reliability of the drive, a sensorless speed control method is proposed. This method allows replacing the physical sensor while guaranteeing a satisfactory operation even in faulty conditions. The novelty of the proposed approach consists of an FPGA implementation of an emulator to control the actual system. Hence, this emulator operates in real-time with actual system control in healthy or faulty mode. It gives an observation of the speed rotation in case of fault for the sake of continuity of service. The observation of the rotor position and the speed are achieved using the dSPACE DS52030D digital platform with a digital signal processor (DSP) associated with a Xilinx FPGA.

This is an open access article under the CC BY-SA license.

Corresponding Author:
Khaldoune Sahri
Laboratoire Commandes des Machines Electriques
Ecole Militaire Polytechnique
P.O. Box 17, 16111 Bordj El Bahri, Algeria
Email: khaldoune.sahri@gmail.com

1. INTRODUCTION

Nowadays we can observe very intensive development of AC variable speed and torque drives. Frequently, their topologies use an association of multi-converters and multi-machines systems. These drives operate in several embedded systems such as vehicles, aircraft, ships, trains, and other industrial applications where weight and volume reduction are essential.

To assure the best quality of PMSM vector control, the rotation speed or the rotor position have to be precisely known. Generally, the principal system variables, such as the currents and rotation speed, are measured using physical sensors. To increase reliability and reduce costs, mechanical sensors have to be suppressed. Consequently, the speed information must be reconstituted from measures of electrical quantities using deterministic state observers. Hence, we reconstruct one or several non-measurable or non-accessible system states, in fact, an observer is an estimator which operates in closed loop and has its own dynamics according to the real system. These dynamics are modified by a gain matrix to cancel the estimation error and ensure convergence. To observe the speed and / or position of a PMSM at different speed levels is always a challenge. For PMSM drive two modes of operation must be taken into consideration of course when operating without mechanical sensors [1]-[3]: i) high and medium speed operation, ii) low speed operation.

Journal homepage: http://ijpeds.iaescore.com
Using the current measurement and the direct current (DC) bus voltage, various algorithms are used to observe the speed of the machine but they do not show satisfactory fonctionnement in a wide range of speed of the system drive. For example, observers who use the return of the electromagnetic field (EMF) and when the machine is driven with insufficient speed to produce a measurable EMF (low speed), in this case the current sensors cannot detect the position of the rotor which cause a bad observation of the speed [1], [4], [5]. Another point to report is that most estimation and observation algorithms of the speed presented in the literature, use an adjustable model with currents measure. In case of a current sensor fault these algorithms become useless.

In addition, the sensitivity of these observers to the parametric variations [5]-[7], and the static converter dead time (inverter) limits the proper functioning of the observer’s and hence the system drive. A dual PMSM system have more degrees of freedom than a single one, (structural actuator and sensors redundancy). So, when a sensor fault appears, a solution that appears very simple is to switch to the actual speed sensor [8]-[10]. The main challenge is to obtain a high dynamic sensorless-drive operation. Taking into account the physical limitations in the real time control of the sensorless synchronous speed drive, we achieve excellent dynamic performances at high and low speeds. The authors interest in this paper, at an experiments of a new control strategy without a speed sensor and without the need to use current sensors it means we observe the rotation speed using the virtual current obtained from the emulator. For a dual PMSM system fed by a dual VSI. We apply an emulator for each machine with its load torque observer and the reference voltages required for the operation of the emulator it means Two individual real time vector controls are applied to drive this global system.

In this way, the obtained deterministic observer replaces the real physical measures, necessary to real-time control system, by corresponding observed variable. Two possibilities exist for working with dSPACE DS52030D digital platform; this emulator function can be implemented on the processor part or on the FPGA target [11]-[13]. To respect its tolerant fault operating two Xilinx FPGA emulators are introduced to increase the system reliability and to admit total physical measure suppression. Certainly, the natural and structural redundancy exists in the DPMSM/DVSI system permitting to switch the speed sensor from first PMSM to second one, or vice versa, when a fault occurs (offset and gain). In the case of current sensor fault its time sharing operation is impossible and it is necessary to dispose three sensors for each PMSM to choose two of them in vector control strategy.

The emulator concept with load torque definition, with "virtual mechanical coupling" in the case of common loads, corresponds to an introduction of analytic redundancy in parallel to real system. In this way the virtual reconstituted variables replace the real physical measures in the control of this system [14]. So, the permanent monitoring detects the sensor failure and generates the decision signal to replace the real measure by corresponding virtual one. The experimental validation of this tolerant sensor fault vector control is made on Laplace DPMSM/DVSI experimental bench and confirms its successful operation and its satisfactory reliability. We present firstly, the different methods for the speed observation that exist in the literature. In the second part, we will put the light on the proposed method which is valid only in the case of a multi-machine system. The studied system is shown in Figure 1.

**Figure 1. General synopsis of studied system**
2. SYSTEM PRESENTATION

2.1. PMSM modelling

The electrical equations expressed in d,q Park frame can be written as [15]:

\[
\begin{align*}
V_d &= R_s I_d + L_s \frac{dI_d}{dt} - \omega_r L_s I_q \\
V_q &= R_s I_q + L_s \frac{dI_q}{dt} + \omega_r L_s I_d + \omega_r \phi_f \\
d\omega_r \over dt &= \frac{P}{J} (L_d - L_q) I_d I_q + \frac{P}{J} \phi_f I_q - \frac{B}{J} \omega_r - \frac{1}{J} T_L
\end{align*}
\]

(1)

Where \( I_d \) and \( I_q \) are the direct-axis and quadrature-axis currents (A), \( V_d \) and \( V_q \) are the direct-axis and quadrature-axis voltage (V), \( \omega_r \) is the velocity of the rotor of the motor (rad/s), \( T_L \) is the load torque (N-m).

We will investigate the control problem of the PMSM with smooth air gap; we will let \( L_d = L_q = L \).

In our study we chose a d,q vector control strategy based on a proportional-integral (PI) cascade control for each PMSM. Thus, the q axis control is made of two loops. The outer loop is for speed control and the inner loop is for the control of the \( I_q \) component. The d axis control has only one loop allowing the direct current regulation \( I_d \).

2.2. State observer

The equations detailed the adjustable model can be rewritten as [16]:

\[
\begin{align*}
(R_s + s L) \dot{I}_d &= \dot{\omega}_r L \dot{I}_q + \dot{\varepsilon}_d + \dot{V}_d + L \dot{l}_{11} \left( I_d - \dot{I}_d \right) + L \dot{l}_{12} \left( I_q - \dot{I}_q \right) \\
(R_s + s L) \dot{I}_q &= \dot{\omega}_r L \dot{I}_d - \dot{\varepsilon}_q + \dot{V}_q + L \dot{l}_{21} \left( I_d - \dot{I}_d \right) + L \dot{l}_{22} \left( I_q - \dot{I}_q \right) \\
s \dot{\varepsilon}_d &= L \dot{l}_{31} \left( I_d - \dot{I}_d \right) \\
s \dot{\varepsilon}_q &= L \dot{l}_{42} \left( I_q - \dot{I}_q \right)
\end{align*}
\]

(2)

where \( \dot{I}_d \) and \( \dot{I}_q \) are the direct-axis and quadrature-axis estimated currents (A), \( \dot{V}_d \) and \( \dot{V}_q \) are the direct-axis and quadrature-axis estimated voltage (V), \( \dot{\omega}_r \) is the observed velocity of the rotor of the motor (rad/s), \( l_{11}, l_{12}, l_{21}, l_{22}, l_{31}, l_{42} \) are adjustable gains used to ensure a good stability of the model. This adjustable model Figure 2 will be used for all the observer versions proposed in this paper such as:

- Rotation speed observation with EMF observation
- Observation of the speed with control of the cross current error.

\[ \text{Figure 2. Adjustable model structure} \]
The following section describe the experimental validation of the two method for the observation of
the rotation speed of the PMSM.

3. EXPERIMENTAL VALIDATION
The parameters of the PMSM are being as:
- Stator resistance $R_s = 2.281\text{ohms}$
- Direct and quadrature inductance $L_d = 23.173\text{mH}$
- Magnetic Flux $\phi_f = 0.241\text{Wb}$
- Number of pairs of poles $P = 4$
- PMSM inertia $J = 22.10^{-4}\text{kg.m}^2$
- Viscous friction coefficient $f = 1.10^{-4}\text{Nms/rad}$

3.1. Experimental bench
This experimental bench is composed of dual PMSM speed controlled drives fed by dual two level
PWM VSIs connected in parallel to DC bus as shown in Figures 3 dan 4. The vector control strategy associated
with sinusoidal SVPWM is chosen to control this experimental bench. The implementation is illustrated in
Figure 5. The common or separated load torque for each studied PMSM is generated thanks to industrial
PMSM controlled torque drives. The torque references are given by the DS1005 processor. In the FPGA target,
the following parts have been implemented:
- ADC, DAC and the two incremental encoders.
- Space Vector PWM strategy (SVPWM).
- Direct and reverse transformations 3-phases $123/\alpha\beta$ and $\alpha\beta/dq$ for the two PMSMs.
- Computing the estimated voltages from the duty cycle of pulses and the DC bus voltage.

Real time emulator for parallel connected dual-PMSM sensorless control (Khaldoune Sahri)
3.2. Observation of the rotation speed with EMF observer status

The Figure 6 shows the principle of the observation of the rotation speed based on the electromagnetic field (EMF) observer status. The EMF $e_d$ is imposed to zero and this quantity is obtained from the adjustable system model Figure 2. Next, the obtained control error must be linearized to assure a good observation of the rotational speed $\hat{\omega}_r$ [16].

![Figure 6. Rotation speed observation with EMF observer status](image)

- Experimental results

In the Figure 7 the rotation speed responses of two M1 and M2 PMSM. The transient and steady state response performances are satisfactory during all speed reference cycles and under different load torques. The responses of dual-PMSM electrical variables, i.e. $I_d$, $I_q$, $V_d$, and $V_q$, are presented in Figures 8 and 9.

![Figure 7. Rotation speed control](image)
3.3. Observation of the speed control by controlling the cross currents error (analytical redundancy)

We can observe the position or the speed thanks to a PI controller using the crossed error between the measured and the estimated currents Figure 10. The principle of this observer is given in the Figure 10. The error is obtained by evaluating:

\[ (I_{qest} \cdot I_d - I_{dest} \cdot I_q) - (I_{qest} - I_q) \cdot \frac{d}{L} \]  

(3)

A robust and stable correction can be based on approach MRAS synthesized using through Popov’s hyper stability criteria [17], [18].

Experimental results

We analyze the PMSM system behavior with the same rotation speed reference cycles. So, the speed responses of M1 and M2 PMSMs are satisfactory during all reference cycles, small oscillations are observed in the steady state after load torque application as shown in Figures 11 and 12. The actual and observed mechanical positions of PMSM are shown in the Figure 13. We notice the effective precision in both rotation directions. In order to test the robustness of the Dual PMSM speed observation, both velocity and load profile
were applied using the different proposed methods. For a wide operation range (low, medium and high speed), these algorithms are not satisfying. To address this issue, each algorithm must be used in one speed range. EMF based observers are not used at low speed because of the low EMF and are better in medium and high speed. Observation of the speed by controlling the cross currents error, is used for low and medium speed range. Several others approaches are used for speed estimation. These methods require huge memory and involve high computational complexity and memory storage [15], [19].

3.4. Natural structural redundancy
In the Dual PMSM system, there is a natural physical redundancy, because each machine has the mechanical or electrical sensors to achieve a closed loop control. This redundancy will be called structural with natural redundancy. The principle of this method is explained on the following figure: The Figure 13 show the proper functioning of the system with physical speed sensor, and when fault sensor operation occurs, we can continue satisfactory operating with only one speed sensor shared by the two machine to achieve a closed loop control with satisfactory operation.
The following method was adopted in order to validate our idea. The state of the signal select change in function of fault in speed sensor. Without fault; the signal select takes the value of zero and when we have fault in the first sensor the signal select passes to 1, then it passes to 2 when we have fault in the second sensor (Figure 14). In our case the decision on faulty speed sensor if the sensor give a value of 80% for the reference speed. For the decisional organ the following algorithms has been implemented:

\[
select = \begin{cases} 
1 & \text{if speed } M1 \geq 80 \% \text{ of ref speed.} \\
2 & \text{if speed } M1 < 80 \% \text{ of ref speed.}
\end{cases}
\]

![Decision device](image)

**Figure 14. Decision device**

- Experimental results

As shown in the Figures 15 and 16, we start the operation with the physical speed sensors that is to say that the system has not yet suffered a defect, then a defect is caused, in our case it is indeed a defect in the second sensor of Speed of the second motor and then we continuing the operation of the system with a single speed sensor shared for the closed loop vector control of the two motors. It should be noted that the natural switching between the physical sensor does not allow the system to be started (it is not possible to start with one speed sensor).

![Rotation speed control](image)

**Figure 15. Rotation speed control**

![Currents Id and Iq](image)

**Figure 16. Currents Id and Iq**

*Real time emulator for parallel connected dual-PMSM sensorless control (Khaldoune Sahri)*
4. THE PROPOSED SENSELESS SPEED FOR THE SYSTEM

We describe in this section the design, implementation, and experimental validation of a real-time hardware-in-the-loop emulation of a dual PMSM. The PMSM machine are modelled using a flexible piecewise linear state space approach [20]; and are simulated in hard real-time with 10 ns time step, which enables high-fidelity modelling of PMSM dynamics. We validate the real-time PMSM drive emulator by making real-time comparisons with a reference hardware model of a PMSM drive. The validation of this real-time DPMSM emulator is made by real-time comparison with a reference hardware model of a DPMSM drive. In this way we want analyze experimentally the operation of the HIL-DPMSM drive emulation under different operating scenarios, in health and faulty conditions. Therefore we experimentally demonstrate the capability of the hardware in the loop PMSM drive emulation under various operating scenarios, including fault conditions [21]-[23].

The implementation of the different control configurations is done using the dSPACE DS52030D digital platform containing a PowerPC DS1005 and a Xilinx FPGA DS5203. The task dispatching between PPC and FPFA can be modified during the final optimized final implementation of global DPMSM drive system and also its load in common or separated configurations. The computing time on the FPGA processor is 10 ns. Thus, we can use the implementation of the continuous mathematical model of the PMSM directly on the FPGA Target. The computing time on the PPC DS1005 is 100 µs. So the mathematical model of the PMSM should be discretized.

In the FPGA target, the following parts have been implemented Figure 17:

- PWM, ADC, DAC and the two incremental encoders.
- Modulation (PWM).
- \(d,q/\alpha\beta\) transformations for the two PMSM.
- \(123/\alpha\beta\) and \(\alpha\beta/d,q\) transformations for the two PMSM.
- Mathematical model of the PMSM in this way a more complex PMSM model can be implemented on FPGA target.

![Figure 17. Different parts of real time control implemented in FPGA and PPC card](image)

All the parts implemented are resumed in Figure 17. The mathematical model of the PMSM Figures 18 requires Luenberger observer to reconstruct the torque load. For the supply voltages \(V_d\) and \(V_q\) necessary for the proper functioning of the emulator, it can be obtained from the references voltages of the actual system. We can obtain them using the DC bus voltage and the pulses of the inverters [24]:

\[
V_{dq} = [d]_dq V_{dc}
\]
\[ I_{dq} = \frac{3}{2} \left[ d \right] d I_{dc} \]
\[ d_d V_{dc} = V_d \]
\[ d_q V_{dc} = V_q \]

Figure 18. The mathematical model of the PMSM

d: is the matrix of control. Unfortunately, if we have a fault in physical sensors the references voltage calculated by this method becomes instable and we do not have enough time to replace the faulty sensor by the emulator variables. To fix this problem, the reference voltages are obtained by a closed loop vector control of the mathematical model implemented on FPGA or PPC in Figure 19. So the voltage are calculated in closed loop and we obtain a robust and reliable observer. For the torque load TL it can be observed by the following algorithm (Figure 20).

Figure 19. Closed loop vector control of the mathematical model

Figure 20. Observed torque load
Experimental results

The model of the PMSM machine is implemented on the FPGA target in order to validate it, in real time conditions. Firstly, the laws are used for the controls implemented on the card PPC DS1005, PWM, and calibration of inputs/outputs. The results presented in Figures 21-24 allow the comparison of the measurable variables (stator currents and speed) of the PMSM, with the ones of the model running, in real time, on the FPGA. This approach validates the proper functioning of the model, the parameter identification and the vector control of the machine. Next, this model is implemented in the FPGA, in order to use the emulated model for the control of the actual system. We validated the real-time emulation by making steady state and transient operation comparisons with the actual PMSM.

Additionally, we demonstrate the ability of the real-time PMSM emulation, as a hardware in the loop prototyping platform, to control the actual system in a wide speed range; from low to high speed. The key advantages of the proposed the real-time emulation platform are twofold. First, the piecewise linear state space modelling approach enables comprehensive modelling of PMSM, including the dynamics that occur on the control [25], [26]. Second, the flexible modelling implementation enables the platform to be used as a powerful tool for rapid prototyping and validation [8]. In the next section, we will present the results obtained during a fault in the speed sensor, the observed torque variable must be reconstructed from the healthy machine as shown in Figures 25-28.

![Figure 21. Comparison of actual and emulated speed](image1)

![Figure 22. Actual current of the PMSM and their emulators](image2)
Figure 23. Actual currents $I_q$ of the PMSM and their emulator

Figure 24. Emulated and actual position

– Experimental results
– Using the emulator FPGA as observer

Figure 25. Rotation speed control

Figure 26. $I_q$ and $I_d$ currents

– Using the emulator PPC as the observer

Real time emulator for parallel connected dual-PMSM sensorless control (Khaldoune Sahri)
The results obtained show clearly that the mathematical model of the machine replaces accurately the physical speed sensor. So, when we impose the rotation speed cycle for this DPMSM we observe good performances of real rotation speeds controlled with emulated one. The load torque modification don’t modify the system precision, the zooms shown in the Figures 16-18 proves the satisfying behavior of this DPMSM system.

The results show that it is possible to continue the operation of the system in the case of a current sensor fault, because the currents \( I_d \) and \( I_q \) required for the vector control can be obtained directly from the emulator which is implemented on the FPGA or the PPC. The results obtained also show that the emulator can be used for the diagnosis and detection of the various existing mechanical fault by the use of the different method of frequency analysis. It means that we can used the emulated mathematical model as a reference system and when we have a mechanical fault, we can easily see this fault in the actual currents by comparing with the emulated currents. Consequently, we can easily conclude on the nature of fault.

The Figure 29 as shown presents a comparison between the reconstitution of speed on FPGA and processor. The results of comparison show that the implemented emulator on the FPGA or the PPC has the same dynamics; so we can distribute the tasks efficiently on FPGA and processor. Therefore, the resources of FPGA can be minimized in a very effective way.
5. CONCLUSION

FPGA is used for the control of two PMSMs with operation at variable speeds and variable torque loads. The objective of this work is to validate the performance of the real time dual-PMSM drive by using the vector control law. We compare the reference system (which is the actual physical system) with the emulated system implemented on FPGA and on the PPC. We also demonstrate the ability of the Hardware-in-the-loop emulation in real time to control the real system under fault of speed sensor conditions and we will demonstrate also other faulty condition for example by injecting open phase fault, and current sensor fault in the emulator drive system. In fact, this emulator can be considered as an adjustable model that does not use gains to adjust the mathematical model; but it uses much more the observation of the load torque as parameter for the adjustment of the model contrary to the existing methods in the literature.

ACKNOWLEDGEMENT

We thank all those who contributed to make this work, including my supervisors in the EMP school and the LAPLACE laboratory in toulouse University and all my friends for their support.

REFERENCES


Real time emulator for parallel connected dual-PMSM sensorless control (Khaldoune Sahri)


