Real-Time Sensorless Control of PMSM and SCADA Integration

Claudiu-Ionel NICOLA[1,2]
[2]Department of Automatic Control and Electronics
University of Craiova
Craiova, Romania
claudiu@automation.ucv.ro

Marcel NICOLA[1]
[1]Research Department
National Institute for Research, Development and Testing in Electrical Engineering-ICMET Craiova
Craiova, Romania
marcel_nicola@yahoo.com

Maria-Cristina NIȚU[1,3]
[3]Department Electrical Engineering, Energetic and Aeronautics
University of Craiova
Craiova, Romania
cristinamarianitu@yahoo.com

2021 International Conference on Applied and Theoretical Electricity (ICATE) | 978-1-7281-8035-9/20/$31.00 ©2021 IEEE | DOI: 10.1109/ICATE49685.2021.9464993

***Abstract*—This article presents an application of real-time control of a Permanent Magnet Synchronous Motor (PMSM) and its integration into Supervisory Control And Data Acquisition (SCADA). Starting from the operating equations of the PMSM and by implementing the global Field Oriented Control (FOC) control strategy, in which the saturation of the integral component of the PI-type controller is prevented by using an anti-windup technique, the numerical simulations performed in Matlab/Simulink lead to good performance, which recommends the real-time implementation. The numerical simulation step is necessary to choose the type of Digital Signal Processing (DSP) used for the real-time implementation, considering that a global criterion of successful implementation is the performance/cost ratio. Moreover, the integration into SCADA provides flexibility of the control system but also the possibility of online/offline processing from the point of view of other specific requirements. Among them we mention the energy quality analysis, whose first exponent calculated also in real-time is Total Harmonic Distortion (THD). Real-time implementations are performed in Matlab/Simulink and LabVIEW programming environments. According to the trend of the last years, the use of an Internet of Things (IoT) platform for viewing the variables of the control process on the Internet has a special place.**

Keywords—permanent magnet motors, embedded system, real-time, SCADA, internet of things

I. INTRODUCTION

The extent of research on PMSM control with or without speed/position encoder is well-known, considering that PMSMs have a number of constructive advantages that meet the requirements of precision electrical actuators, robotics, and computer peripherals [1-5]. Among the many types of control we can mention from the PI-type controllers, to the adaptive, predictive and intelligent control type of controllers [6-9].

Further, one of the most widely used speed observers is the SMO-type observer. The Matlab/Simulink environment is commonly used to compare the performance of these control systems. Thus, high-precision numerical simulations can be performed, which can reveal the advantages and disadvantages of various types of controllers. This analysis is inherently followed by the real-time implementation of the control system in embedded systems. The designer of these systems must weigh in the balance both the performance of the control systems studied by numerical simulations, and the cost of their real-time implementations in embedded systems. The real-time implementations are performed in Matlab/Simulink and LabVIEW programming environments [10-12].

Moreover, there is an inherent problem of creating a local control interface, as well as the problem of integration in the local SCADA. Thus, the control can be performed via secure communications from the Intranet/Internet. The integration into SCADA provides the transmission of signals of interest to other client systems. Thus, on the one hand, the computer server is no longer forced to perform a series of processing tasks, and on the other hand, there is the possibility of running such tasks online or offline, followed by the efficient management of global computing resources. In this sense, a separate task can run online on a client computer to monitor the quality electrical parameters of the controlled PMSM. The main parameter analyzed, but not the only one is the THD. For increased flexibility, according to the trend of the last years, the use of an IoT platform for viewing the variables of the PMSM control process on the Internet has a special place [13-17].

The presented article can be considered as a follow-up on a series of articles presented by the same authors regarding the numerical simulations of the PMSM control using various types of controllers. Thus, it presents the real-time implementation of the PMSM control and its integration into SCADA.

The rest of the paper is structured as follows: Section II describes the mathematical model and the numerical simulation for the sensorless control of the PMSM. The real-time implementation of the sensorless control of the PMSM and its integration into SCADA is presented in Section III. Section IV presents the experimental results of the control system. The final section presents some conclusions and some ideas for the next papers.

II. THE MATHEMATICAL MODEL AND THE NUMERICAL SIMULATION FOR THE SENSORLESS CONTROL OF THE PMSM.

The mathematical model of the PMSM in the d-q reference frame is the following [1], [6-8]:

$$\begin{bmatrix} u_q \\ u_d \end{bmatrix} = \begin{bmatrix} R_q + \rho L_q & \omega_e L_d \\ -\omega_e L_q & R_d + \rho L_d \end{bmatrix} \begin{bmatrix} i_q \\ i_d \end{bmatrix} + \begin{bmatrix} \omega_e \lambda_0 \\ \rho \lambda_0 \end{bmatrix} \quad (1)$$

where: u_d, u_q - stator voltages; R_d, R_q - d and q axis resistances; L_q, L_d - d and q axis inductances; i_d, i_q - d anq q axis currents; ω_{and} - electrical angular velocity of the rotor; λ_0 - flux linkage; ρ - differential operator.

The flux of the PMSM engine is expressed by the following relations:

The paper was developed with funds from the Ministry of Education and Scientific Research - Romania as part of the NUCLEU Program: PN 19 38 01 03 and PN 19 38 02 04.

$$\lambda_q = L_q i_q$$
$$\lambda_d = L_d i_d + \lambda_0 \quad (2)$$

The following relations are obtained by using the notation T_e for the electromagnetic torque developed by the PMSM:

$$T_e = \frac{3}{2} n_p \left(\lambda_d i_q - \lambda_q i_d\right)$$
$$T_e = K_t i_q \quad (3)$$
$$T_e = T_L + B\omega + J\frac{d\omega}{dt}$$

where: $K_t = \frac{3}{2}(n_p \lambda_0)$ - torque constant; B - viscous friction coefficient; J - rotor inertia; np - number of pole pairs; TL - the load torque.

For $L_d = L_q = L$, $R_d = R_q = R_s$, and $\omega_e = n_p \cdot \omega$, where ω is the angular velocity of the rotor, the following PMSM model can be obtained:

$$\begin{pmatrix} \dot{i}_d \\ \dot{i}_q \\ \dot{\omega} \end{pmatrix} = \begin{pmatrix} -\frac{R_s}{L} & n_p\omega & 0 \\ -n_p\omega & -\frac{R_s}{L} & -\frac{n_p\lambda_0}{L} \\ 0 & \frac{K_t}{J} & -\frac{B}{J} \end{pmatrix} \begin{pmatrix} i_d \\ i_q \\ \omega \end{pmatrix} + \begin{pmatrix} \frac{u_d}{L} \\ \frac{u_q}{L} \\ -\frac{T_L}{J} \end{pmatrix} \quad (4)$$

Figure 1 shows the general diagram for the PMSM sensorless control. The control system is based on the global FOC strategy. It can be noted that the anti-windup PI controller is fed at the input with the error between the reference speed and the speed estimated by the SMO-type observer and generates i_{qref} at the output. The stator currents are acquired from the PMSM windings, and currents i_d and i_q are obtained by applying the transformation in the d-q reference frame. The errors between these currents and their reference values i_{qref} and i_{dref} (i_{dref} is set to zero according to the FOC strategy) are inputs for two PI controllers which in turn generate the voltages u_d and u_q.

By using the inverse Park transform, as can be noted in Figure 1, the currents in the α-β reference frame are obtained and expressed in the following form [1], [6]:

$$\frac{di_\alpha}{dt} = -\frac{R_s}{L} i_\alpha - \frac{1}{L} e_\alpha + \frac{1}{L} u_\alpha$$
$$\frac{di_\beta}{dt} = -\frac{R_s}{L} i_\beta - \frac{1}{L} e_\beta + \frac{1}{L} u_\beta \quad (5)$$

The equations of the SMO-type observer for the estimation of the PMSM rotor speed and position are based on the estimation of the back-EMF e_α and e_β.

$$\frac{d\hat{i}_\alpha}{dt} = -\frac{R_s}{L}\hat{i}_\alpha + \frac{1}{L}u_\alpha - \frac{1}{L}kH(\hat{i}_\alpha - i_\alpha)$$
$$\frac{d\hat{i}_\beta}{dt} = -\frac{R_s}{L}\hat{i}_\beta + \frac{1}{L}u_\beta - \frac{1}{L}kH(\hat{i}_\beta - i_\beta) \quad (6)$$

Where the k parameter represents the observer gain, and function H is of sigmoid type:

$$H(x-y) = \frac{2}{1+e^{-a(x-y)}} - 1 \quad (7)$$

where: a represents a positive constant, and the sigmoid function indicated in relation (7) will be assigned values between -1 and 1 for a = 4.

Based on these, the estimates of the back-EMF are obtained in the following form [6]:

$$\hat{e}_\alpha = kH(\bar{i}_\alpha) = -\lambda_0 \hat{\omega}_e \sin\theta_e$$
$$\hat{e}_\beta = kH(\bar{i}_\beta) = \lambda_0 \hat{\omega}_e \cos\theta_e \quad (8)$$

The PMSM rotor speed and position estimates expressed in relations (9) and (10) are obtained from the relation (8).

$$\hat{\omega}_e = \frac{\sqrt{\hat{e}_\alpha^2 + \hat{e}_\beta^2}}{\lambda_0} \quad (9)$$

$$\hat{\theta}_e(t) = \int_{t_0}^{t} \hat{\omega}_e(t)dt + \theta_0 \quad (10)$$

where: θ_0 is the initial electrical position of the rotor.

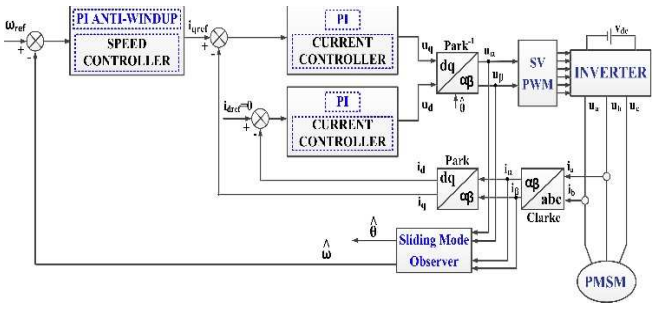

Fig. 1. The general diagram for the PMSM sensorless control based on the FOC strategy.

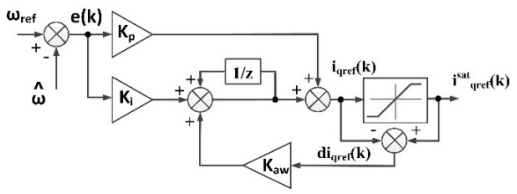

Fig. 2. The block diagram of the PI anti-windup speed controller.

The discrete time equation of the PI-type controller with anti-windup is the following:

$$i_{qref}(k) = \left[K_p + \left(K_i + di_{qref}(k)K_{aw}\right)\frac{T_s z}{z-1}\right]e(k) \quad (11)$$

Figure 3 shows the evolution in time of the estimated speed of the PMSM, of the load torque and electromagnetic torque, of the stator currents, and currents i_d and i_q obtained based on the numerical simulation performed in Matlab/Simulink.

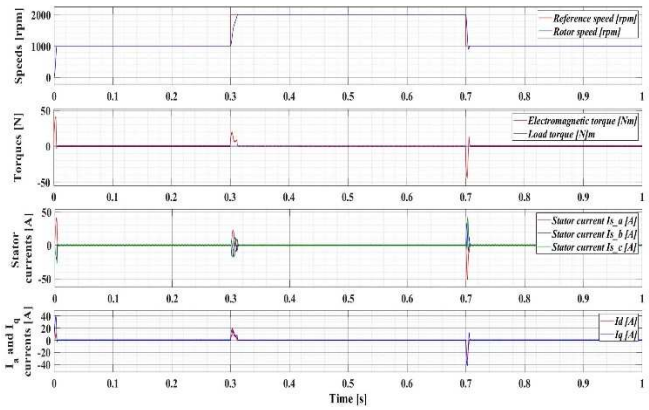

Fig. 3. Time evolution for the numerical simulation of the PMSM sensorless control system.

TABLE I. NOMINAL PARAMETERS OF THE PMSM

Parameter	Value	Unit
Rated voltage	24	F
Rated power	55	W
Rated speed	4000	rpm
Stator winding resistance - R_s	0.405	Ω
Stator winding inductance - L_s	0.63e-3	H
Rotor inertia - J	4.6e-6	kg·m^2
Rotor friction - B	1.13e-6	N·m·s/rad
Permanent magnet flux linkage - λ_0	0.175	WB
Current at rated speed	5	a)
Torque at rated speed	0.185	[Nm]
Back-EMF constant - k_{and}	0.0172	V/rad/s
Pole pairs number - P	4	-

III. THE ARCHITECTURE FOR THE REAL-TIME SENSORLESS CONTROL OF THE PMSM AND SCADA INTEGRATION

Figure 4 shows the proposed architecture for the integration into SCADA of the real-time control system of a PMSM. The characteristics and operating equations of the PMSM, together with the control possibilities are presented in the previous section by means of numerical simulations in Matlab/Simulink.

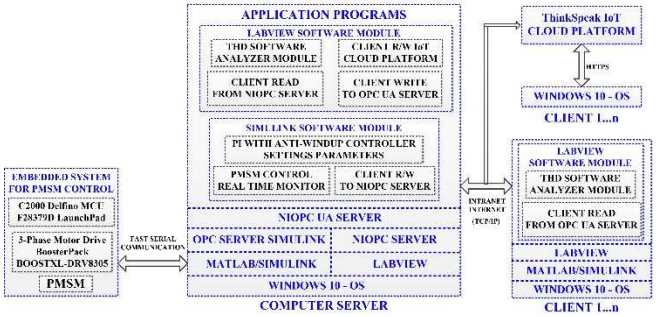

Fig. 4. The proposed general diagram for SCADA integration of the PMSM sensorless control system.

For the real-time implementation, we will use a development platform consisting of LAUNCHXL-F28379D and three-phase drive stage BOOSTXL DRV8305EVM. LAUNCHXL-F28379D use USB connected isolated XDS100v2 JTAG debug probe for real-time debug and flash programming; and contain TMS320F28379D MCU with the following hardware characteristics: 200 MHz dual C28xCPUs and dual CLAs; 1 MB Flash; 16-bit/12-bit ADCs; 12-bit DACs; comparators; filters; HRPWMs; eCAPs; eQEPs; CANs. The BOOSTXL-DRV8305EVM is based on the DRV8305 motor gate driver and CSD18540Q5B power MOSFET. This type of module has individual DC bus and phase voltage sense as individual low-side current shunt sense for sensorless algorithms used for the control of the PMSM [15].

The block diagram of the software application for the real-time control of the PMSM is implemented in Simulink and is shown in Figure 5. By analogy to Figure 1, we note the implementation of the FOC type strategy, of which we mention the outer control loop for the control of speed (see Figure 6), the inner control loop for the control of currents i_d and i_q (see Figure 7), and the communication loop which will be required later in the software application for monitoring the control process (see Figure 8). The running time of the two control loops is 0.05ms for the current loop, and for the speed loop it is 0.5ms. The setting parameters for the current controllers are K_p=2.78 and K_i=50074.6, and for the speed controller are K_p=0.92 and K_i=24.35.

The usual software blocks for real-time implementation are provided by the Motor Control Blockset Toolbox (Park and Clarke transforms, the sensorless observers, the anti-windup IP, etc.). By means of the Embedded Coder Support Package for TI C2000 Processors, the application program is translated into C language or machine code to be downloaded to the F28379D-type microcontroller. Additionally, a code optimization is performed, of which we mention the translation from fixed point to adjustable point, resulting in major improvements of the control performances in real-time [15].

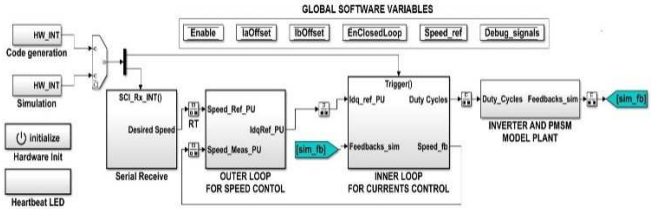

Fig. 5. The block diagram of the software application for real-time implementation in the embedded system.

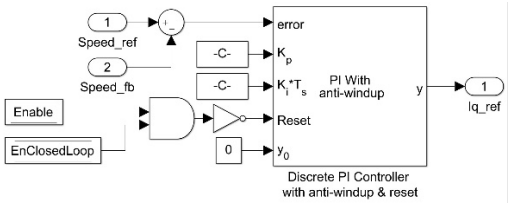

Fig. 6. The block diagram for implementation in Matlab/Simulink of the speed controller.

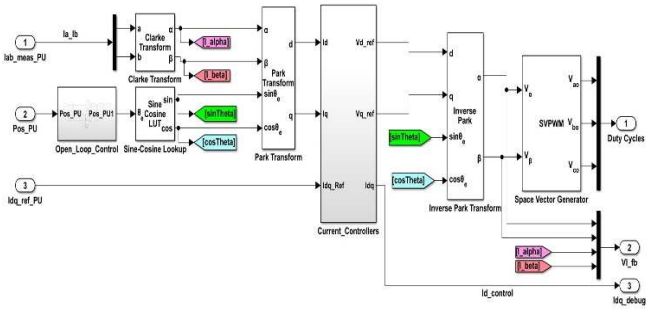

Fig. 7. The block diagram for implementation in Matlab/Simulink of the current controllers and PWM generation signals.

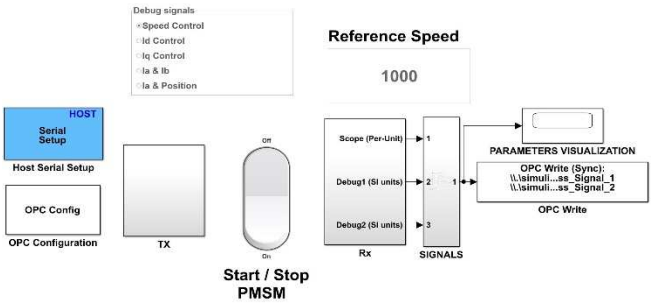

Fig. 8. The block diagram of the software application for monitoring the control process of the PMSM.

The data stream for supervising the PMSM control between Matlab/Simulink and LabVIEW is achieved through the connection between the OPC servers embedded in these software development environments. For this purpose, we define the network-published shared variable, by which the data in the Intranet network based on Ethernet TCP/IP can be written and read. The Shared Variable Engine (EVS) mechanism uses the NI Publish-Subscribe Protocol (NI-PSP) to transfer data corresponding to the variables distributed in the data network [13, 16].

Figure 9 shows the connection achieved between the OPC server embedded in Simulink and the "*localhost/National Instruments.Variable Engine.1*" server embedded in LabVIEW using blocks from the OPC Toolbox Simulink. The actual writing of the PMSM control process variables is performed by means of the OPC Write block in SVE and is presented in Figure 10. These variables are accessible in software programs developed in LabVIEW on the computer server, but are also accessible to the embedded OPC UA server to achieve the communication with client computers connected to the Intranet/Internet.

Figure 11 shows the LabVIEW project tree with the software applications developed on the computer server and the SVE library.

Figure 12, 13 and 14 show the OPC UA server configuration tasks and the read/write tasks which ensure communication between the OPC UA servers on the computer server and the client software applications on the computers connected to the Internet.

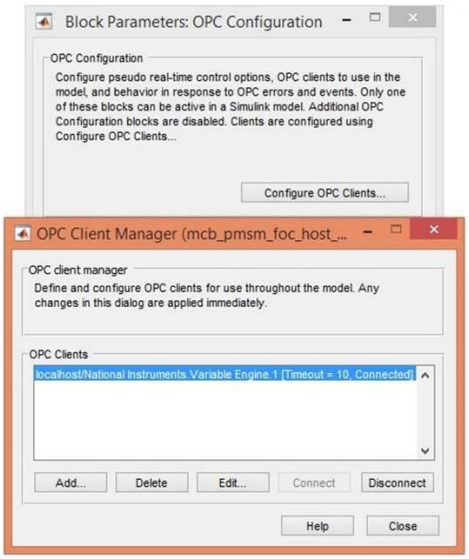

Fig. 9. Simulink OPC Server configuration.

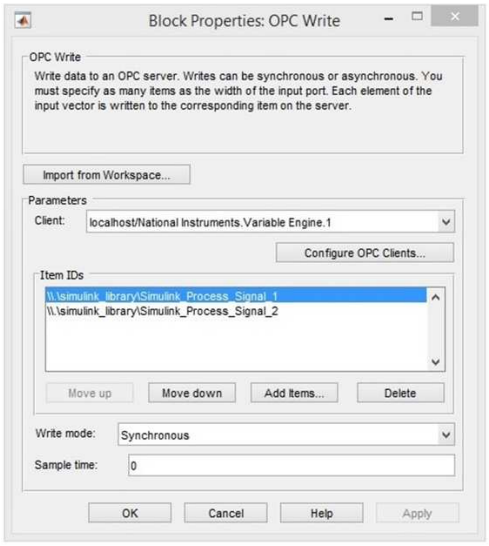

Fig. 10. Simulink OPC Server write variables of the control process.

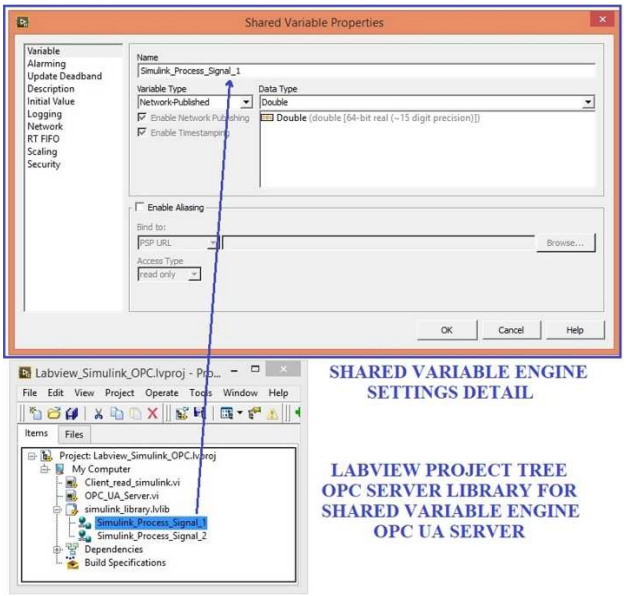

Fig. 11. LabVIEW project tree for the software applications on the computer server.

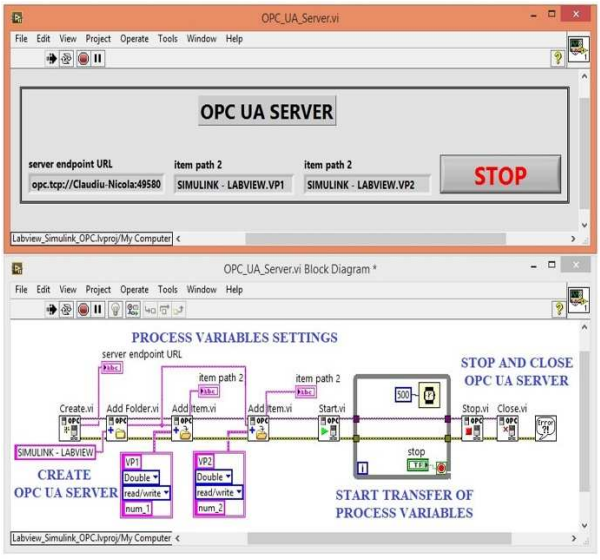

Fig. 12. OPC UA server configuration and implementation in LabVIEW.

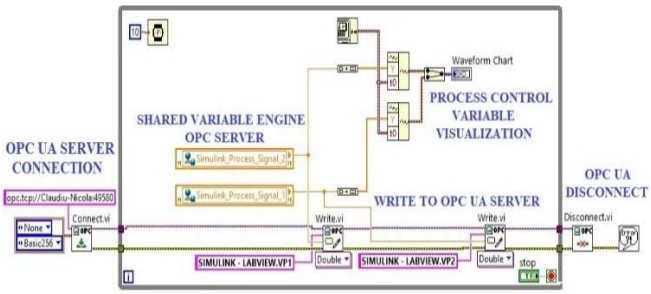

Fig. 13. Implementation in LabVIEW of the OPC UA client write.

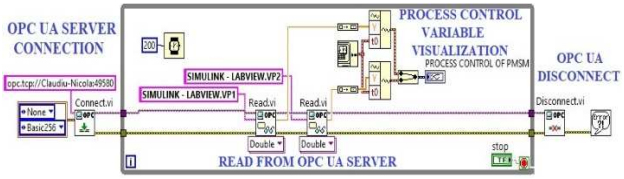

Fig. 14. Implementation in LabVIEW of the OPC UA client read.

Further, a separate task can run online on a client computer to monitor the quality electrical parameters of the controlled PMSM [14]. Figure 15 shows the implementation in a client/LabVIEW for THD analysis.

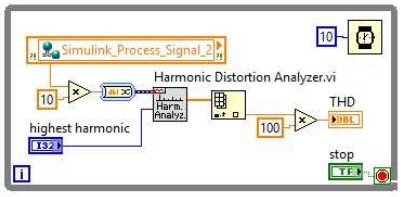

Fig. 15. Implementation in a client/LabVIEW of the task for the THD analysis.

For greater flexibility in sharing data from the PMSM control process, an IoT platform can also be used. Thus, the viewing, analysis and processing of data in the cloud is performed. The platform can be accessed from both Matlab/Simulink and LabVIEW using specific GET, POST, PUT, and DELETE commands. These commands create, write, read, or delete communication channels. These commands are HTTP requests and responses and fit into the representational state transfer (REST) architecture [17]. The configuration of a communication channel used in the presented application can be done at the address "*https://thingspeak.com/channels/1306529*" and is presented schematically as in Figure 16.

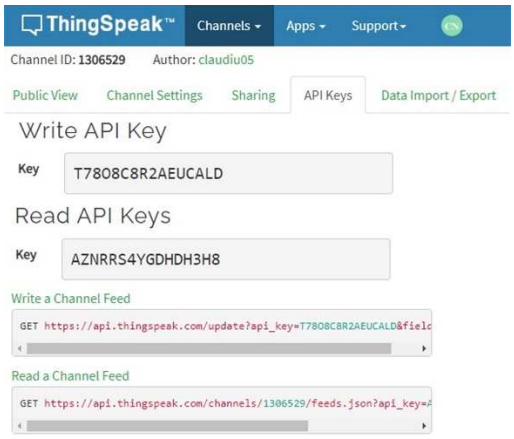

Fig. 16. Configuration settings of the IoT server for read / write channel.

Writing data to this channel is done from a separate task implemented in LabVIEW on the computer server (see Figure 17).

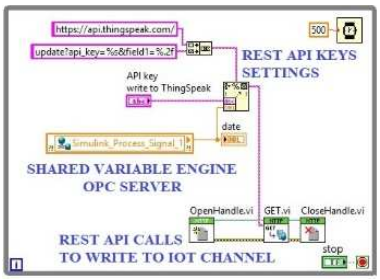

Fig. 17. Configuration settings of the IoT server for read / write channel.

IV. Experimetal Results

The PMSM sensorless control system was tested in real-time in two stages. The loggings of the speed and the parameters of interest (currents and torque) are presented in the first stage using the interface on the host computer/computer server (see Figures 18-22). In the second stage, the integration into SCADA and the communication with IoT cloud platform were tested (see Figures 23 and 24).

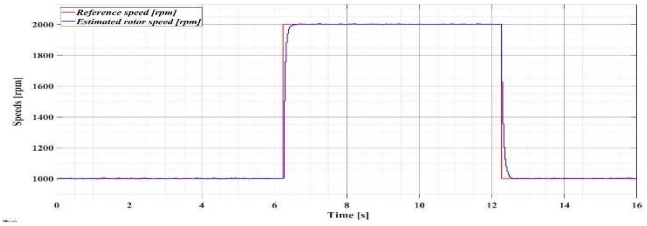

Fig. 18. Time evolution of the estimated PMSM rotor speed.

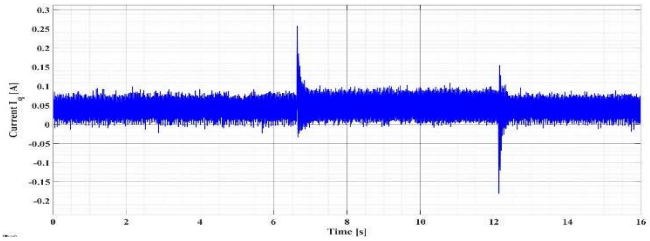

Fig. 19. Time evolution of current I_q.

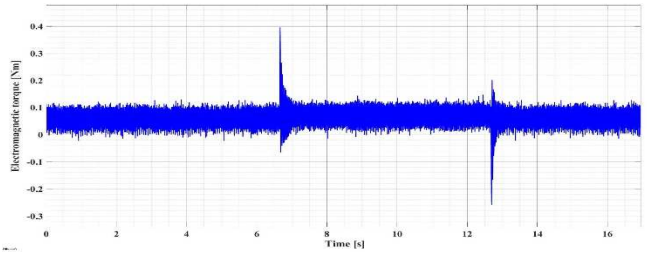

Fig. 20. Time evolution of the electromagnetic torque T_e.

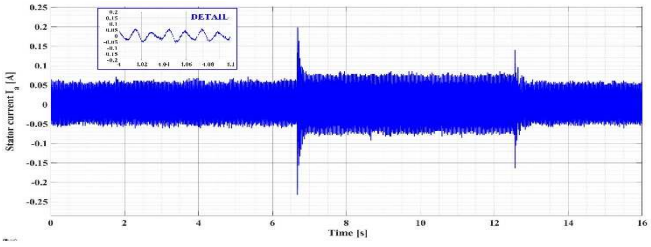

Fig. 21. Time evolution of the stator current I_a.

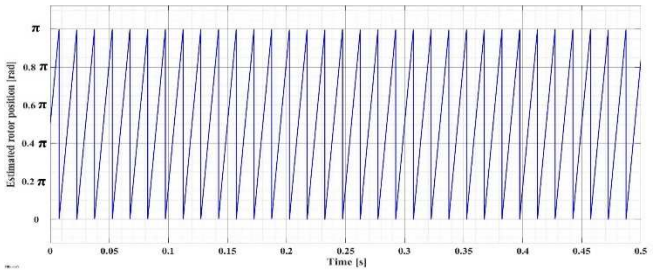

Fig. 22. Time evolution of the PMSM rotor position.

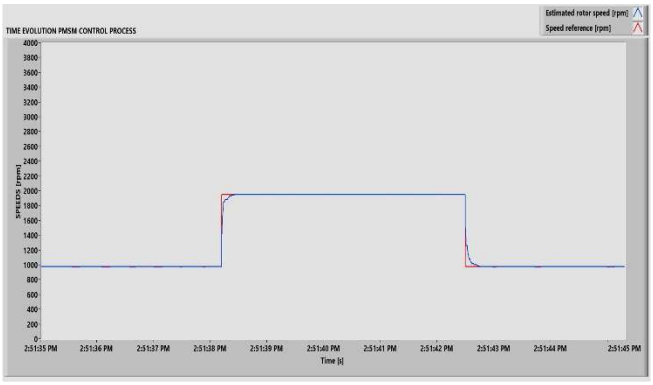

Fig. 23. Time evolution of the estimated PMSM rotor speed on OPC UA client read.

Fig. 24. Time evolution of the estimated PMSM rotor speed on IoT cloud platform.

V. Conclusions

This article presents the mathematical models of a PMSM, of the FOC-type sensorless control, using a PI-type control with anti-windup and an SMO-type rotor speed and position observer. It presents both the numerical simulations and the real-time implementation of the control system using a LAUNCHXL-F28379D-type development and control platform and three-phase drive stage DRV8305EVM from Texas Instruments.

The integration into SCADA is achieved using OPC-type servers. For greater flexibility, in the sense of using computers with no Simulink or LabVIEW software development modules installed, an IoT platform is used. In future papers we will focus on the real-time implementation of a multi-motor application in which the control of parameters is performed locally or on the Intranet through proper integration into the local SCADA system.

Acknowledgment

This paper was developed partially in Grant POCU380 / 6/13/123990, co-financed by the European Social Fund within the Sectorial Operational Program Human Capital 2014–2020.

References

[1] V. Utkin, J. Guldner, J. Shi, Sliding mode control in electromechanical systems, second edition. Automation and Control Engineering, Taylor & Francis, 2009.

[2] B. K. Bose, Modern power electronics and AC drives, Prentice Hall, Knoxville, Tennessee, USA, 2002.

[3] H. Wang and J. Leng, "Summary on development of permanent magnet synchronous motor," *Chinese Control And Decision Conference (CCDC)*, Shenyang, China, 2018, pp. 689-693.

[4] Z. Liu, Y. Li, and Z. Zheng, "A review of drive techniques for multiphase machines," in CES Transactions on Electrical Machines and Systems, vol. 2, no. 2, pp. 243-251, June 2018.

[5] S. Sakunthala, R. Kiranmayi, and P. N. Mandadi, "A Review on Speed Control of Permanent Magnet Synchronous Motor Drive Using Different Control Techniques," *International Conference on Power, Energy, Control and Transmission Systems (ICPECTS)*, Chennai, China, 2018, pp. 97-102.

[6] M. Nicola and C. I. Nicola, "Sensorless Control for PMSM Using Model Reference Adaptive Control and back -EMF Sliding Mode Observer," *International Conference on Electromechanical and Energy Systems (SIELMEN)*, Craiova, Romania, 2019, pp. 1-7.

[7] M. Nicola and C. I. Nicola, "Sensorless Predictive Control for PMSM Using MRAS Observer," *International Conference on Electromechanical and Energy Systems (SIELMEN)*, Craiova, Romania, 2019, pp. 1-7.

[8] M. Nicola, C. I. Nicola, and M. Duță, "Sensorless Control of PMSM using FOC Strategy Based on Multiple ANN and Load Torque Observer," *International Conference on Development and Application Systems (DAS)*, Suceava, Romania, 2020, pp. 32-37.

[9] M. Mutluer and O. Bilgin, "Design optimization of PMSM by particle swarm optimization and genetic algorithm," *International Symposium on Innovations in Intelligent Systems and Applications*, Trabzon, Turkey, 2012, pp. 1-4.

[10] C. I. Nicola, M. Nicola, A. Vintilă, and D. Sacerdoțianu, "Identification and Sensorless Control of PMSM Using FOC Strategy and Implementation in Embedded System," *International Conference and Exhibition on Electromechanical and Energy Systems (SIELMEN)*, Craiova, Romania, 10-11 October, 2019, pp. 335-340.

[11] Z. Chen, "Permanent Magnet Synchronous Motor Parameter Test System Based on LabVIEW," *3rd Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, Chongqing, China, 2019, pp. 434-437.

[12] M. Sreejeth, M. Singh, P. Kumar, P. Varshney, and P. Sachdeva, "Implementation of supervisory control system for PMSM drive," *5th India International Conference on Power Electronics (IICPE)*, Delhi, India, 2012, pp. 1-6.

[13] M. Nicola, C. I. Nicola, M. Duță, D. Sacerdoțianu, "SCADA Systems Architecture Based on OPC and Web Servers and Integration of Applications for Industrial Process Control," in International Journal of Control Science and Engineering, vol. 8, no. 1, pp. 13-21, April 2018.

[14] C. I. Nicola *et al*., "Quality analysis of electric energy using an interface developed in LabVIEW environment," *International Conference on Applied and Theoretical Electricity (ICATE)*, Craiova, Romania, 2016, pp. 1-6.

[15] Motor Control Blockset™ User's Guide, Matlab and Simulink, MathWorks, Natick, MA, 2020.

[16] M. S. Mahmoud, M. Sabih, M. Elshafei, "Using OPC technology to support the study of advanced process control," in ISA Transactions, vol. 55, pp. 155-167, March 2015.

[17] The IoT Platform with MATLAB Analytics. Available [online]: https://nl.mathworks.com/help/thingspeak/