

FOC with Resolver Implementation for PMSM Drives by Using a Low Cost Atmel SAM3X8E Microcontroller

M. Caruso, A.O. Di Tommaso, R. Miceli, *Member IEEE*, C. Nevoloso, G. Vassallo and G. Vitamia

Department of Engineering, University of Palermo
Viale delle Scienze, Building nr. 9, 90128 - Palermo, Italy
antoninooscar.ditommaso@unipa.it

Abstract—The aim of this paper is the low-cost experimental implementation of a field oriented control strategy for a Permanent Magnet Synchronous Motor (PMSM) by using an Atmel SAM3X8E microcontroller, mounted on an Arduino DUE board. In this electrical drive for PMSM, a resolver is used in order to measure the rotor position and speed: therefore, the low-cost Arduino DUE performs not only FOC algorithm and phase currents data acquisition, but also a resolver-to-digital converter process, rotor position and speed data acquisition, and resolver signals management. The code has been implemented in the open source Arduino IDE, using C language, whereas the control and plot visualization interfaces have been developed by using an open source software. In order to make currents and resolver conditioning signals, a double-layer printed circuit board (PCB) has been created. The electrical drive has been experimentally tested and the main results reported in this work show high performance in terms of rotor speed response and compilation and execution speed of the code, validating the use of Arduino DUE for PMSM drives.

Index Terms—Permanent magnet synchronous motor (PMSM), electrical drives, microcontroller, Field oriented Control (FOC)

NOMENCLATURE

θ_{ref} [rad]= reference of rotor mechanical angle;
 θ_m [rad]= rotor mechanical angle;
 θ_e [rad]= rotor electrical angle;
 $ia(t)$ [A]= instantaneous value of phase A current;
 $ib(t)$ [A]= instantaneous value of phase B current;
 $ic(t)$ [A]= instantaneous value of phase C current;
 $i\alpha$ [A]= phase A current referred to α axis;
 $i\beta$ [A]= phase B current referred to β axis;
 id_{ref} [A]= reference of current direct component;
 iq_{ref} [A]= reference of current quadrature component;
 id [A]= current direct component;
 iq [A]= current quadrature component;
 $va(t)$ [V]= instantaneous value of phase A voltage;
 $vb(t)$ [V]= instantaneous value of phase B voltage;
 $vc(t)$ [V]= instantaneous value of phase C voltage;
 vd_{ref} [V]= reference of voltage direct component;
 vq_{ref} [V]= reference of voltage quadrature component;
 va_{ref} [V]= reference of voltage component referred to α axis;
 $v\beta_{ref}$ [V]= reference of voltage component referred to β axis;
 v_{sin} [V]= sine voltage signal, resolver output;
 v_{cos} [V]= cosine voltage signal, resolver output;
 dn = duty cycle for phase n ;
 U_{DC} [V]= DC-link voltage value;
 u_{n-rij} [V] = reference voltage value, for phase n .

I. INTRODUCTION

PMSM electrical drives are increasingly widespread, due to distinctive features that characterize these machines. Although a higher cost than other motors, caused by the presence of permanent magnets, they have a great robustness and reliability, because of the absence of rotor windings and above all the absence of brushed system for excitation current injection. Furthermore, PMSMs are suitable for several application fields, such as automotive applications and high precision robotics due to their compactness and high torque/inertia ratio [1]-[18].

The high cost of PMSMs falls on the cost of the whole electric drive. Therefore, the aim of the research is to identify control devices, such as microprocessors as well as microcontrollers, less expensive, in order to reduce the overall cost of the system, but maintaining the same high performance. In industrial or automotive applications, systems such as DSP (digital signal processing), FPGA (field programmable gate array) or CPLD (complex programmable logic device) are often used; however, in spite of their high operating frequency, and the resulting high performance, their prices could reach high values (see Table I) [17]–[26].

TABLE I. ECONOMIC ANALYSIS

Controller	Model	Frequency	Price
Arduino	Arduino DUE	84 MHz	\$ 38.00
DSP	C2000 controlCARDs-TMS320F2808	100 MHz	\$ 61.00
DSP	ADZS-BFSHUSB-EZEXT	400 MHz	\$ 112.72
DSP	TMDSCNCD28335 controlcard	150 MHz	\$ 69.00
DSP	TMDSPREX28335 controlCARD	150 MHz	\$ 195.00
DSP	TMDSDOCK28069	90 MHz	\$109.00
FPGA +CPLD	Xilinx® Inc. EK-S6-SP605-G	1.1 GHz	\$ 783.21
FPGA	Xilinx® Spartan®-6 LX4 Digilent Cmod s6	133 MHz	\$ 86.00
FPGA	Xilinx® Spartan®-7 Digilent Arty S7-25	50 MHz	\$ 99.00
FPGA	Zynq-7000 Xilinx® 7 Cora Z7	667 MHz	\$ 99.00
FPGA	DE0-Nano - Altera Cyclone IV	50 MHz	\$ 99.95
Max V CPLD	Intel DK-DEV-5M570ZN	247 MHz	\$ 74.9

In this paper, the Authors will show the application of the Atmel SAM3X8E microcontroller, mounted on Arduino DUE, for PMSM electrical drives, instead of traditional controllers. The most important feature of this device is the low cost, i.e., at present days, 35 €, with a high performance control but with a considerably reduced budget compared to other systems (Table I). Moreover, the Arduino DUE board provides a higher flexibility, since the interface pins are already available and immediately usable, working at lower clock frequencies – 84 MHz – compared to other systems, but more than enough for electrical drives applications [27, 28].

The system is characterized by a high compilation and execution speed of the code, implemented in a programming C language, very widespread in industrial field; moreover, only 5% of the memory is occupied by the code, so it can be developed to implement other features, e.g. faults management (fault protection), as well as implementation of loss minimization algorithms. The code is developed in Arduino IDE, an open source program, and written in fixed point algebra, in order to improve microcontroller performance. Furthermore, another open source program, i.e. *Processing*, is used to create the user interface with the front control panel for the electric drive, in order to make graphic data acquisition, for post processing in the MATLAB or equivalent environments.

In addition, the electric drive carried out consists of a surface permanent magnet synchronous motor (SPMSM), Mavilor BL-073, a DPS-30A converter with IGBT bridge and a resolver Tamagawa Smartsyn 2T8, speed and position transducer. The Arduino DUE board controls the drive, manages the resolver input and output signals, making also data acquisition, for phase currents and for sine and cosine signals of the resolver. In order to adapt the acquired signals to the limit values of the Arduino DUE pins, conditioning circuits are needed and, for this goal, a double layer PCB has been created in the Rapid Prototyping Laboratory (RPLAB) [32].

Therefore, this paper presents the low-cost experimental implementation of a PMSM field oriented control strategy by using a low-cost microcontroller. The related electrical drive has been experimentally tested and the obtained results show that the dynamic performances of the drive are comparable with those obtained by microcontrollers with higher costs.

II. PMSM FOC AND RDC ALGORITHMS

A. PMSM Field Oriented Control

The block diagram of the PMSM FOC is shown in Fig.1. The decoupling between magnetization flux and torque is the most important achievement that characterizes this control strategy, increasing the machine control performance, if compared to other algorithms. In this control logic, the direct-axis current component i_d is set equal to 0, without field-weakening algorithm, in order to get a control simplification and to avoid the demagnetization of the permanent magnets.

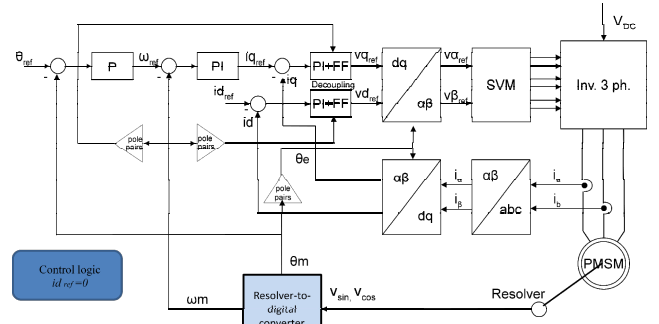


Fig.1 FOC block diagram.

Clarke transformation ($abc \rightarrow \alpha\beta$) and Park transformation ($\alpha\beta \rightarrow dq$) are needed in order to change from a stationary three-phase reference frame to a rotating two-phase one:

$$\begin{aligned} i_\alpha &= i_a \\ i_\beta &= \frac{1}{\sqrt{3}}i_a + \frac{2}{\sqrt{3}}i_b \end{aligned} \quad (1)$$

$$\begin{aligned} i_d &= i_\alpha \cos(\theta_e) + i_\beta \sin(\theta_e) \\ i_q &= -i_\alpha \sin(\theta_e) + i_\beta \cos(\theta_e) \end{aligned} \quad (2)$$

The inverse transformations are given by:

$$\begin{aligned} v_\alpha &= v_d \cos(-\theta_e) + i_q \sin(-\theta_e) \\ v_\beta &= -v_d \sin(-\theta_e) + i_q \cos(-\theta_e) \end{aligned} \quad (3)$$

$$\begin{aligned} v_a &= v_\alpha \\ v_b &= -\frac{1}{2}v_\alpha + \frac{\sqrt{3}}{2}v_\beta \\ v_c &= -(v_a + v_b) \end{aligned} \quad (4)$$

A sine function is implemented through a look-up table, in order to compute these transformations. The sine function has 12-bit of resolution, so its values change from 0 to 4095 (maximum value) and it has 4096 samples in a period. The cosine function is calculated as follows:

$$\cos(\theta_e) = \sin(\theta_e + 1024) \quad (5)$$

in order to considerer 90° phase shift between the functions.

B. Space vector pulse width modulation (SVPWM)

For this electric drive the space vector pulse width modulation has been chosen. In this algorithm, the duty cycle is directly calculated for each inverter phase leg, considering a distinction between the linear modulation region,

$$d_n = \frac{1}{2} + \frac{u_{n-rif} + U^*}{U_{DC}} \quad (6)$$

in which

$$U^* = -\frac{1}{2} \{ \min(u_{a-r}, u_{b-rif}, u_{c-rif}) + \max(u_{a-rif}, u_{b-rif}, u_{c-rif}) \} \quad (7)$$

and the over-modulation region, for which the following formula is used:

$$d_n = \frac{1}{2} + \frac{u_{n-rif} + U^*}{U^{**}} \quad (8)$$

where

$$U^{**} = \{ \max(u_{a-rif}, u_{b-rif}, u_{c-rif}) - \min(u_{a-rif}, u_{b-rif}, u_{c-rif}) \} \quad (9)$$

In order to verify if the system is in the linear or in the over-modulation region, a geometric approach is adopted: by making the intersection between a rhombus and a rectangular strip, with specific dimensions, a hexagon is obtained, whose area is expressed by the following inequalities:

$$\begin{cases} \sqrt{3}|u_{r\alpha}| + |u_{r\beta}| \leq \frac{2}{\sqrt{3}} U_{DC} \\ |u_{r\beta}| \leq \frac{U_{DC}}{\sqrt{3}} \end{cases} \quad (10)$$

If these expressions are fulfilled, the reference vector is contained within the hexagon and, therefore, a linear modulation occurs [32]. Otherwise, (8) and (9) are used. PWM counters resolution is 12-bit, so U_{DC} is equal to 4096 in the Arduino software.

C. Resolver-to-Digital Converter (RDC)

Rotor and position transducer that has been chosen for this electric drive is the resolver, which has a greater robustness against vibrations, high temperature and electrical noise than other measuring systems, e.g. encoder, providing also high reliability, while maintaining good performance in terms of measurement accuracy. Generally, the rotor position and speed are calculated using a specific Integrated Circuit (IC), implementing a Resolver-to-Digital Converter (RDC). In this case, the microcontroller implements directly the RDC, achieving high accuracy of the rotor position identification, with a very low cost and with high performance. The analog part of the RDC is characterized by a limited number of components, which increases the transducer reliability and reduces its price, as well as EMI interferences and thermal problems [32]. The RDC contains only one PI controller, leading to a simplification of the software implementation.

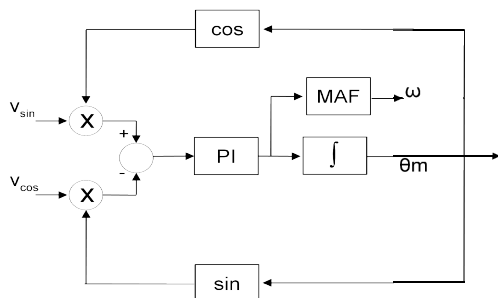


Fig.2 Resolver-to-Digital Converter diagram block.

The scheme of fig.2 shows the RDC software logic [32]. In this application, the frequency of the resolver excitation signal, as well as the resolver output signals, is equal to 10

kHz. The 10 kHz sine and cosine modulated outputs, elaborated by a conditioning circuit in the range of 0-3.28V, are multiplied by $\sin\phi$ and $\cos\phi$, in which ϕ is an estimated value of the rotor position. Then, the error between these products is computed by using their difference. This error is processed by a PI regulator and the resulting signal is filtered by a Moving Average Filter (MAF), to get the speed. Finally, the mechanical position is achieved by integration over time [32].

III. SOFTWARE AND HARDWARE REALIZATION

A. Structure of the code

The Arduino code is written by following the fixed-point algebra rules, in order to improve the microcontroller performances in terms of execution speed. The implementation logic is based on the interrupt system. An interrupt is generated by the *Timer Counter* (TC) unit, whereas another by the *PWM* controller. Interrupts are signals generated by a peripheral or a software request, which stop the main loop function, in order to start-call cyclically another instruction, the interrupt functions. The first interrupt (TC), with a higher priority, is synchronized to the PWM channels of Arduino DUE, in order to create the carrier signal to supply the resolver. A sawtooth wave counter is generated by using the TC with the following frequency:

$$f_{pwm} = \frac{MClk}{2 * CPRD3} = \frac{84MHz}{2 * 128} = 328,125 kHz \quad (18)$$

in which $MClk$ is the Masterclock frequency, whose value is equal to 84 MHz, and $CPRDx$ is the channel period register, set to 128 (see Fig. 3) [29]. The sampled reference wave is a sine function implemented in a look-up table, with 32 samples and 7-bit resolution (128 points). On the top of each saw-tooth counter, an interrupt is generated.

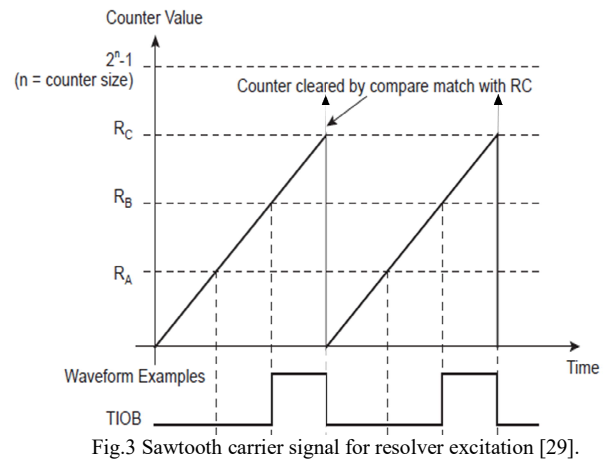


Fig.3 Sawtooth carrier signal for resolver excitation [29].

The PWM frequency (symmetrical modulation) has been consequently chosen from the following formula:

$$f_{signal} = \frac{f_{pwm}}{samples} = \frac{328125}{32} = 10253,9 Hz \quad (19)$$

A conditioning circuit is needed, in order to raise the amplitude of this signal, from 2.8 V to about 10 V, and to filter the curve, getting a sinusoidal shape.

The second interrupt is related to the PWM channels (0,1,2) and data acquisition for ADC, which is 32 times slower than the TC interrupt, with a lower priority. By using registers, a triangular carrier wave is generated, with 4096 points (12-bit resolution) and the following frequency (fig.4):

$$f_{pwm} = \frac{MClk}{2 * CPRDx} = \frac{84MHz}{2 * 4096} = 10253,9 Hz \quad (20)$$

In this case, CPRDx period is also set to 4096 [28, 29]. The 12-bit PWM resolution is compatible with 12-bit sine function stored in the same look-up table of the Clarke-Park transformations; the PWM period is, therefore, equal to $T_{pwm} = 97.5 \mu s$. The PWM interrupt is activated each time the counter goes to zero (see Fig. 4) and a *PWM_handler* function is activated, in which the FOC algorithm is executed, including vector modulation and the RDC. The *PWM_Handler* is called every 97.5 μs , equal to PWM period. The PWM channels are set in *PWM_setup* by means of registers. Furthermore, when the triangular counter reaches the top, another interrupt is sent, to acquire both the phase current signals and the resolver signals (sine and cosine); the ADC channels are set by using the *ADC_setup* registers. Moreover, by taking advantage of the synchronism between the TC and PWM interrupts, the sampling of the resolver outputs occurs always at the same time instant, corresponding to the maximum of the resolver output sine signal. Therefore, the demodulation of the resolver outputs is achieved. There are also other two interrupts, related to UOTGHS and USART0, in order to carry out bidirectional serial communication. Some libraries have been created in order to implement the mathematical functions, PI regulators, lookup tables and files containing the parameters of the electrical drive (motor and converter rated values, etc).

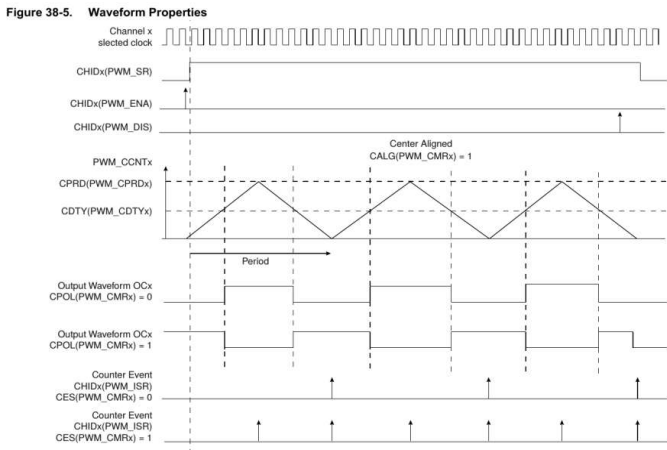


Fig.4 Triangular carrier signal for *PWM_Handler* function [29].

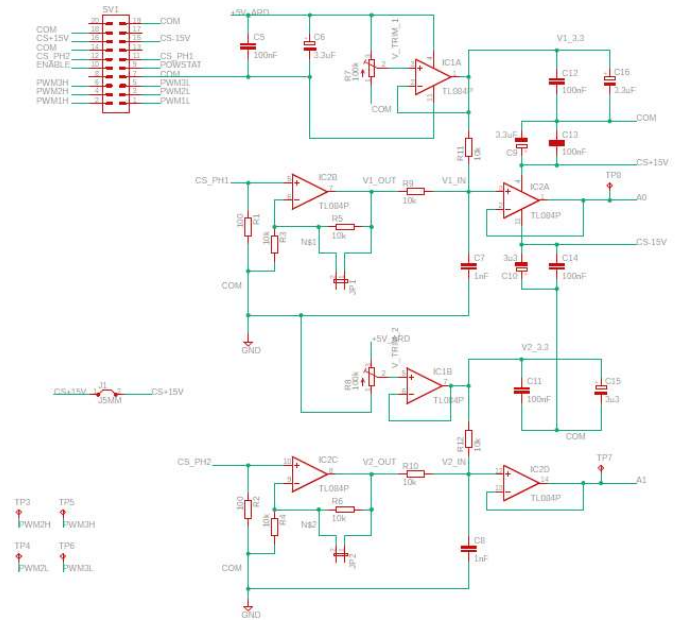


Fig.5 Conditioning circuits for phase currents.

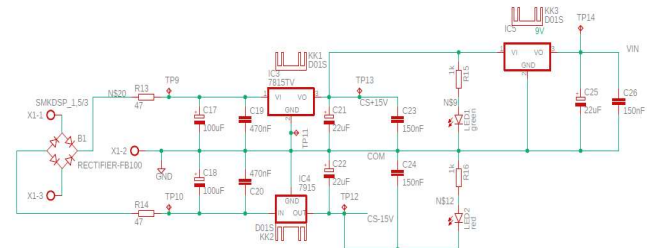


Fig.6 Power supply circuit.

B. Conditioning circuits

A double-layer Printed Circuit Board (PCB) is needed for the conditioning circuits for phase current signals, as well as for resolver signals (input and output).

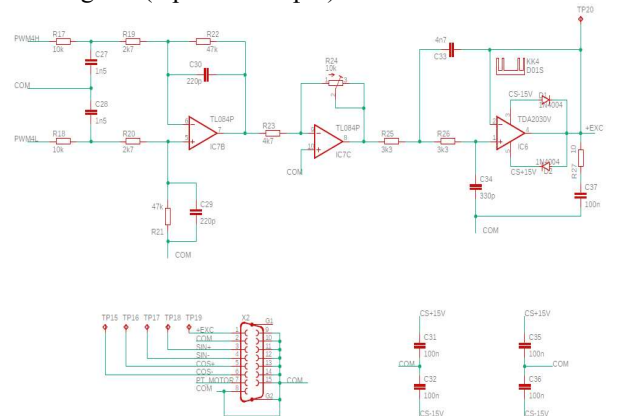


Fig.7 Conditioning circuit for resolver input signal.

All the schematic in this paper are made by means of Autodesk EAGLE. In the current conditioning circuit, measured phase currents are converted in voltage signals and are adapted to the limit value of Arduino DUE pin, that is 3.3V.

The schematic in fig.5 shows this conditioning circuit. For each phase current, a precision resistance is used to convert the current value into voltage signal. There are two stages of the same TL084 operational amplifier. Between the two stages, a RC low-pass filter is implemented and a further voltage follower with a trimmer is used in order to settle the offset for each signal.

Fig.6 shows the electric power supply for the other electronic components (operational amplifiers and Arduino board), with a 30VA transformer with two secondary coils and some voltage regulators, such as 7815 and 7915.

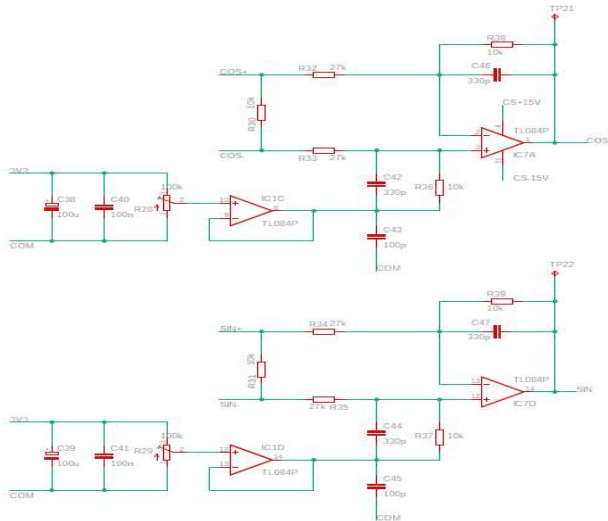


Fig.8 Conditioning circuit for resolver output signals.

The scheme in fig. 7 shows the conditioning circuit for the resolver excitation signal. Two stages of TL084 are implemented in order to obtain:

- 1) an inverting differential operational amplifier, with a specific transfer function to increase voltage value from 2.8V to 10V and to achieve an almost perfect sinusoidal wave.
- 2) a buffer (follower) with a 10 kΩ trimmer (R_{24}) to make voltage adjustment.

Furthermore, there is a Sallen-Key filter, to make an additional low-pass filter step for excitation signal, a power operational amplifier TDA203, used as a power supply for the resolver excitation winding.

Fig. 8 shows the conditioning circuits related to the sine and cosine resolver output signals.

The double-layer PCB with these conditioning circuits has been created in the RPLab at the University of Palermo. The fabrication process consists of drilling, electroplating, milling and painting. The resulting board is shown in fig. 9.

IV. EXPERIMENTAL RESULTS AND DISCUSSIONS

In order to validate the use of the Atmel SAM3X8E for electric drive applications, several experimental tests have been carried out at the Electric Drives Applications Laboratory (EDALab). A Mavilor BLS-073 surface permanent magnet motor has been used and its rated values are summarized in Table II. The power converter used in this FOC

electric drive is the DPS 30A, made by Automotion INC, and characterized by an IGBT bridge (the technical features are reported in Table III). In addition, a Yokogawa DL1740 scope completes the test bench (see fig.10).

Graphic data are sent by the *loop* function implemented in Arduino code to the *Processing* open-source software, used to create user interface and the control panel. This software is developed in JAVA.



Fig. 9 Shield composed of PCB double layer and Arduino DUE board.

TABLE II. MAVILOR BLS-073 SPSM DATA

MAVILOR BLS-073	Value
Rated voltage [V]	220
Rated current [A]	3,91
Nr. of pole pairs	4
Max. mechanical speed [rpm]	11000 rpm
Stall torque [N·m]	2,7
Peak torque [N·m]	16,31
Winding resistance [Ω]	3,2
Direct axis inductance [mH]	3,2
Quadrature axis inductance [mH]	3,2
Voltage constant k_{emf} [V/rad/s]	0,41
Rotor inertia [kg·m ²]	0,000074
Torque/weight ratio [Nm/kg]	1,17
Torque constant k_t [Nm/s]	0,71
Mass [kg]	2,3
Protection	IP-65
Resolver	Tamagawa Smartsyn 2T8

TABLE III. DPS30A DATASHEET

DPS30A CONVERTER AUTOMOTION INC.	Value
Input voltage [V]	230
DC-Link voltage	310
Maximum output power [VA]	6,5
Peak output current [A]	30
Switching frequency [kHz]	5...20 kHz
Current sensors (F.W. Bell)	CLN - 25

Firstly, a no-load test and a load test are carried out. In both cases, a speed reverse is performed, from 550 rad/s to -550 rad/s. Figs.11 and 12 show the currents and the speed during the reverse transient at no load, respectively. Each transient duration lasts only 0.2 s, highlighting the high dynamic response. It can be noted that the steady-state value of i_d is equal to 0 A, according to the chosen FOC. The peak mechanical speed is equal to $\omega=610$ rad/s.

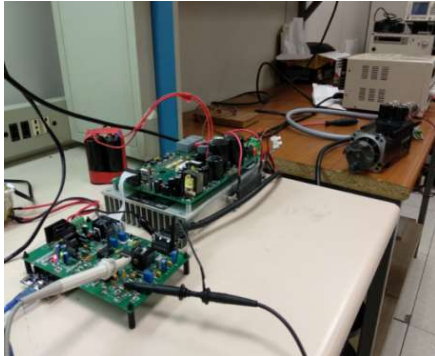


Fig. 10. A photograph of the test bench.

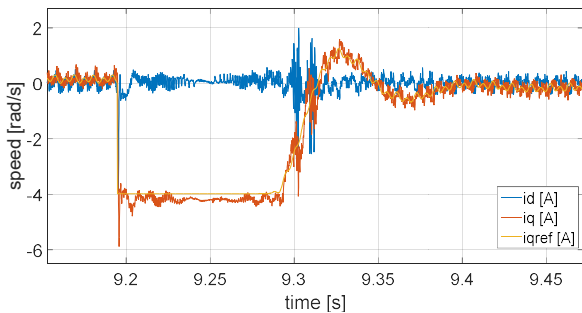


Fig. 11. Currents reverse transients, no load test.

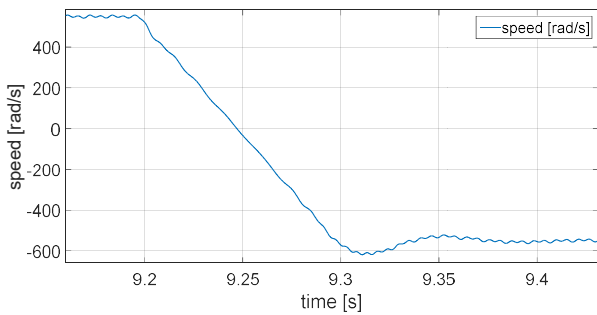


Fig. 12. Speed reverse transient, no load test.

In fig.13 and fig.14, the reverse transients are shown for a load test. A comparison can be carried out between this test and the previous one: transient durations for speed last 0.2 s; however, a difference in terms of peak values and a more oscillating response can be found in this test. The high dynamic response persists even in this case. Peak speed is equal to 600 rad/s, lower than the no load test; moreover, it can be observed that the i_q trend follows the i_q reference signal, demonstrating the good performance of the implemented drive.

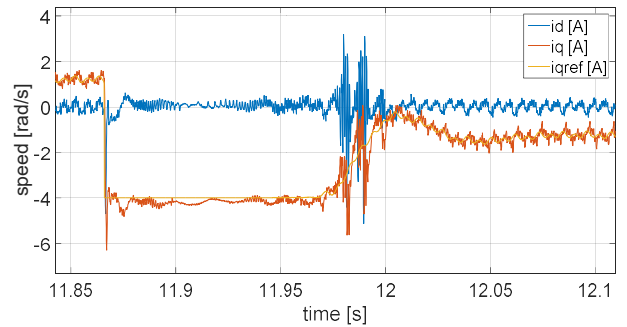


Fig. 13. Current reverse transient, load test.

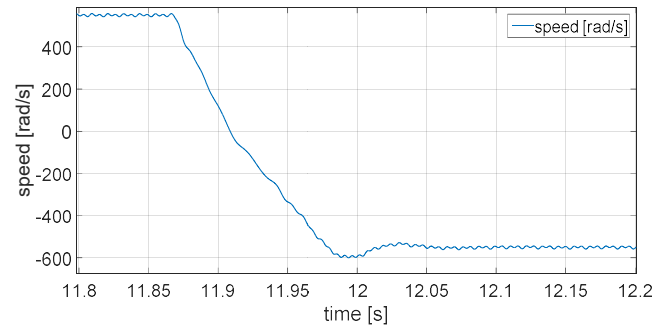


Fig. 14. Speed reverse transient, rated load test.

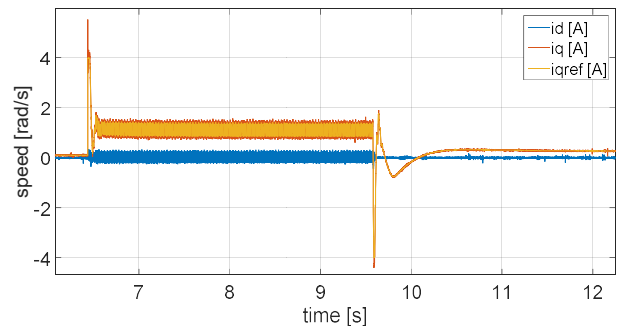


Fig. 15. Position control, currents curves in rated load test.

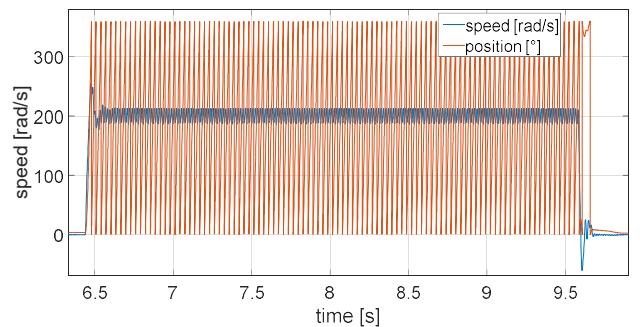


Fig. 16. Position control, speed and position curves.

With regard to the implementation of the position control, the number of rotor revolutions are imposed as reference into the code. The Authors have carried out a load test in which 100 revolutions are set for the rotor; its maximum speed has been fixed to 200 rad/s and the load was the same as in the speed load tests.

Fig. 15 shows the trends of the currents, in which the peak of i_q is equal to 5.8 A and its steady-state value ranges between 1.5 A and 1 A. The i_d has an oscillation around 0 A, coherently to the FOC logic. The initial transient lasts about 0.3 s for both current components. Fig.16 shows the speed and position curves. The speed has a peak value equal to 250 rad/s and steady-state value ranging between 190 rad/s and 210 rad/s. An initial transient of 0.3 s can be observed.

The last test made by the Authors is the measure of *PWM_Handler* and *TC0_Handler* durations, displayed in Yokogawa DL1740 scope, by using Arduino pins nr. 43 and 44, properly set. *PWM_Handler* is executed in 28 μ s, only 29% of the 97 μ s PWM period after which this *handler* function is cyclically called (fig. 17). Instead, the code performing the RDC lasts only 600ns (fig. 18).

Both of these debug measurements show the high execution speed that characterizes the Atmel SAM3X8E microcontroller.



Fig. 17 Debug PWM.

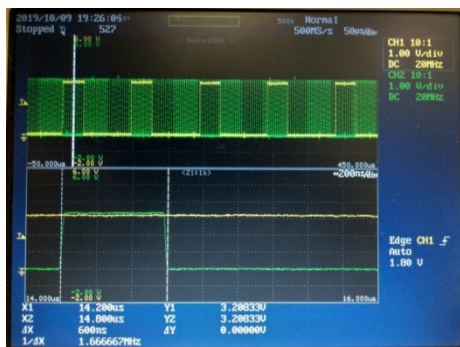


Fig.18 Debug TC0.

V. CONCLUSIONS

This work has presented a low-cost experimental implementation of a FOC strategy in a PMSM drive by using an Atmel SAM3X8E microcontroller.

The code has been implemented in the open source Arduino IDE, using C language, whereas the control and plot visualization interfaces have been developed by using an open source software with the code written in Java.

The experimental results carried out by means of an IPMSM test bench have demonstrated that the proposed drive, even if implemented by means of a low-cost microcontroller, provides high performances and good dynamic response. Finally, the

possibility of modifying the parameters of the code increases the flexibility of the control system, which can be potentially adapted to several applications, such as automotive and industrial ones.

ACKNOWLEDGMENTS

This work was financially supported by PON R&I 2015-2020 “Propulsione e Sistemi Ibridi per velivoli ad ala fissa e rotante – PROSIB”, CUP no:B66C18000290005, by H2020-ECSEL-2017-1-IA-two-stage “first and european sic eightinches pilot line-REACTION”, by Prin 2017- Settore/Ambito di intervento: PE7 linea C - Advanced power-trains and -systems for full electric aircrafts, by PON R&I 2014-2020 - AIM (Attraction and International Mobility), project AIM1851228-1 and by ARS01_00459-PRJ-0052 ADAS+ “Sviluppo di tecnologie e sistemi avanzati per la sicurezza dell'auto mediante piattaforme ADAS”.

REFERENCES

- [1] E. V. Kazmin, E. A. Lomonova and J. J. H. Paulides, "Brushless traction PM machines using commercial drive technology, Part I: Design methodology and motor design," *2008 International Conference on Electrical Machines and Systems*, Wuhan, 2008.
- [2] E. V. Kazmin, E. A. Lomonova and J. J. H. Paulides, "Brushless traction PM machines using commercial drive technology, Part II: Comparative study of the motor configurations," *2008 International Conference on Electrical Machines and Systems*, Wuhan, 2008.
- [3] A. O. di Tommaso, R. Miceli, G. R. Galluzzo and M. Trapanese, "Efficiency Control for Permanent Magnet Synchronous Generators," *2006 IEEE International Conference on Industrial Technology*, Mumbai, 2006.
- [4] Kaiping Yu, Hong Guo, Zedong Sun and Zhiyong Wu, "Efficiency optimization control of Permanent Magnet Synchronous Motor for Electric Propulsion System," *2013 International Conference on Electrical Machines and Systems (ICEMS)*, Busan, 2013, pp. 56-61.
- [5] M. Caruso, A. O. Di Tommaso, M. Lombardo, R. Miceli, C. Nevoloso and C. Spataro, "Experimental comparison of two control algorithms for low-saliency ratio interior permanent magnet synchronous motors," *2018 Thirteenth International Conference on Ecological Vehicles and Renewable Energies (EVER)*, Monte-Carlo, 2018, pp. 1-6.
- [6] M. Caruso, A. O. Di Tommaso, R. Miceli, C. Nevoloso, C. Spataro and F. Viola, "Enhanced loss model algorithm for interior permanent magnet synchronous machines," *2017 AEIT International Annual Conference*, Cagliari, 2017, pp. 1-6.
- [7] M. Caruso, A. O. Di Tommaso, F. Genduso and R. Miceli, "Experimental investigation on high efficiency real-time control algorithms for IPMSMs," *2014 International Conference on Renewable Energy Research and Application (ICRERA)*, Milwaukee, WI, 2014, pp. 974-979.
- [8] C. Cavallaro, A. O. Di Tommaso, R. Miceli, A. Raciti, G. Ricco Galluzzo and M. Trapanese, "Analysis a DSP implementation and experimental validation of a loss minimization algorithm applied to permanent magnet synchronous motor drives," *IECON'03. 29th Annual Conference of the IEEE Industrial Electronics Society (IEEE Cat. No. 03CH37468)*, Roanoke, VA, USA, 2003, pp. 312-317 vol.1.
- [9] E. Simon, "Implementation of a Speed Field Oriented Control of 3-phase PMSM Motor using TMS320F240", Application Report SPRA588, Texas Instruments, 1999.
- [10] A. Cataliotti, F. Genduso, A. Raciti and G. R. Galluzzo, "Generalized PWM-VSI Control Algorithm Based on a Universal

Duty-Cycle Expression: Theoretical Analysis, Simulation Results, and Experimental Validations," in *IEEE Transactions on Industrial Electronics*, vol. 54, no. 3, pp. 1569-1580, June 2007.

[11] F. Pellitteri, V. Boscaino, A. O. Di, R. Miceli, G. Capponi, "Experimental test on a Contactless Power Transfer system", *Ecological Vehicles and Renewable Energies (EVER) 2014 Ninth International Conference on*, pp. 1-6, 2014.

[12] Pellitteri, F., Caruso, M., Castiglia, V., Miceli, R., Spataro, C., Viola, F., "Experimental Investigation on Magnetic Field Effects of IPT for Electric Bikes", (2018) *Electric Power Components and Systems*, 46 (2), pp. 125-134.

[13] Caruso, M., Di Tommaso, A.O., Marignetti, F., Miceli, R., Galluzzo, G.R. A general mathematical formulation for winding layout arrangement of electrical machines (2018) *Energies*, 11 (2), art. no. 446.

[14] Caruso, M., Di Tommaso, A.O., Imburgia, A., Longo, M., Miceli, R., Romano, P., Salvo, G., Schettino, G., Spataro, C., Viola, F. Economic evaluation of PV system for EV charging stations: Comparison between matching maximum orientation and storage system employment (2017) *2016 IEEE International Conference on Renewable Energy Research and Applications, ICRERA 2016*, art. no. 7884519, pp. 1179-1184.

[15] Schettino, G., Buccella, C., Caruso, M., Cecati, C., Castiglia, V., Miceli, R., Viola, F. Overview and experimental analysis of MC SPWM techniques for single-phase five level cascaded H-bridge FPGA controller-based (2016) *IECON Proceedings (Industrial Electronics Conference)*, art. no. 7793351, pp. 4529-4534.

[16] L. Schirone, M. Macellari, F. Pellitteri, "Predictive dead time controller for GaN-based boost converters", *Power Electron.*, vol. 10, no. 4, pp. 421-428, 2017.

[17] F. Pellitteri, M. Caruso, V. Castiglia, A. O. Di Tommaso, R. Miceli, L. Schirone, "An inductive charger for automotive applications", *Industrial Electronics Society IECON 2016 - 42nd Annual Conference of the IEEE*, pp. 4482-4486, 2016.

[18] F. Pellitteri, A. O. Di Tommaso, R. Miceli, "Investigation of inductive coupling solutions for E-bike wireless charging", *Power Engineering Conference (UPEC) 2015 50th International Universities*, pp. 1-6, 2015.

[19] Ming-Fa Tsai and Hsien-Chang Chen, "Design and implementation of a CPLD-based SVPWM ASIC for variable-speed control of AC motor drives," *4th IEEE International Conference on Power Electronics and Drive Systems. IEEE PEDS 2001 - Indonesia. Proceedings (Cat. No.01TH8594)*, Denpasar, Indonesia, 2001.

[20] X. Zhou, H. Zhao and J. Zhu, "Hardware design of the PMSM control system based on DSP and CPLD," *2015 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS)*, Okinawa, 2015.

[21] J. Lin, K. W. E. Cheng, Z. Zhang and X. Xue, "Experimental investigation of in-wheel switched reluctance motor driving system for future electric vehicles," *2009 3rd International Conference on Power Electronics Systems and Applications (PESA)*, Hong Kong, 2009, pp. 1-6.

[22] D. Luczak, "DSP implementation of electric drive control system," *2012 8th International Symposium on Communication Systems, Networks & Digital Signal Processing (CSNDSP)*, Poznan, 2012, pp. 1-3.

[23] L. Huang, "Low cost control of two three-phase PMSMs in integrated motor drives and battery chargers for electric vehicle application," *2015 18th International Conference on Electrical Machines and Systems (ICEMS)*, Pattaya, 2015, pp. 1894-1897.

[24] R. K. Pongiannan and N. Yadaiah, "FPGA based Space Vector PWM Control IC for Three Phase Induction Motor Drive," *2006 IEEE International Conference on Industrial Technology*, Mumbai, 2006, pp. 2061-2066.

[25] M. Ruba, S. Ciornei, H. Hedesiu and C. Martis, "Complete FPGA based real-time motor drive simulator with bidirectional battery and ultracapacitor power supply," *2017 10th International Symposium on Advanced Topics in Electrical Engineering (ATEE)*, Bucharest, 2017, pp. 186-191.

[26] S. Benanti, C. Buccella, M. Caruso, V. Castiglia, C. Cecati, A.O. Di Tommaso, R. Miceli, P. Romano, G. Schettino and F. Viola, "Experimental analysis with FPGA controller-based of MC PWM techniques for three-phase five level cascaded H-bridge for PV applications," *2016 IEEE International Conference on Renewable Energy Research and Applications (ICRERA)*, Birmingham, 2016.

[27] M. Caruso, V. Castiglia, A. Del Pizzo, R. Miceli, M. Salles. G. Schettino, V. Traversa and F. Viola, "Low-cost smart energy management based on ATmega 328P-PU microcontroller," *2017 IEEE 6th International Conference on Renewable Energy Research and Applications (ICRERA)*, San Diego, CA, 2017, pp. 1204-1209.

[28] V. Castiglia, P. Ciotta, A. O. Di Tommaso, R. Miceli and C. Nevoloso, "High Performance FOC for Induction Motors with Low Cost ATSAM3X8E Microcontroller," *2018 7th International Conference on Renewable Energy Research and Applications (ICRERA)*, Paris, 2018, pp. 1495-1500.

[29] "Sam3x / sam3a series atmel smart arm-based mcu", in *DATASHEET*.

[30] Arduino, Official website, in <https://store.arduino.cc/duo>

[31] Processing, official website, in <https://processing.org/>

[32] M. Caruso, A. O. Di Tommaso, F. Genduso, R. Miceli and G. R. Galluzzo, "A DSP-Based Resolver-To-Digital Converter for High-Performance Electrical Drive Applications," in *IEEE Transactions on Industrial Electronics*, vol. 63, no. 7, pp. 4042-4051, July 2016.