BULETINUL INSTITUTULUI POLITEHNIC DIN IAŞI
Publicat de
Universitatea Tehnică „Gheorghe Asachi” din Iaşi
Volumul 62 (66), Numărul 2, 2016
Secţia
ELECTROTEHNICĂ. ENERGETICĂ. ELECTRONICĂ

CURRENT SAMPLING BASED SATURATION LIMITS DESIGN OF PI REGULATORS IN MOTOR CONTROL APPLICATIONS

BY

ALEXANDRA IOSUB[1,*], JEROME KIRSCHER[2], MONICA RAFAILA[2], ARCADIE CRACAN[1], ANDI BUZO[2], GEORG PELZ[2] and LIVIU GORAŞ[1,3]

[1]“Gheorghe Asachi” Technical University of Iaşi
Faculty of Electronics, Telecommunications and Information Technology,
[2]Infineon AG Neubiberg, Germany,
[3]Institute of Computer Science, Romanian Academy, Iaşi

Received: June 1, 2016
Accepted for publication: June 30, 2016

Abstract. The paper discusses an electric bicycle (E-Bike) application developed with SystemC-AMS, a C++ based language for system-level description, model refinements and simulation improvements. The system simulations have showed that the saturation limits of the currents of PI controllers must be adjusted from their theoretical value $V_{\mathrm{Batt}}/\sqrt{3}$ to a new one which depends on the current sensor bandwidth. Besides including this saturation in the duty cycle of the Pulse Width Modulation (PWM) signals, an adjustment of the synchronization between the PWM and the Analog to Digital Converter (ADC) blocks is done by properly generating interrupt events such that the ADCs correctly sample the currents sensed via the shunt resistor based sensors. Besides, we show that it is possible to model, validate and optimize a complex system using only one description language, SystemC-AMS, an open-source C++ library.

Key words: SystemC-AMS; PMSM; E-Bike; PI; current sensor; shunt resistor.

---

[*] Corresponding author: *e-mail*: Alexandra.Iosub-EE@infineon.com

1. Introduction

Permanent magnet (PM) synchronous motor has gained an increasing popularity in applications such as electric vehicles due to its following assets: high torque/inertia ratio, high power density and high efficiency. In this paper we investigate an E-Bike application (Fig. 1). An E-Bike is a bicycle with an integrated electric motor which can be used for propulsion. Modeling such an application implies dealing with heterogeneous simulations in terms of signal type (digital and analog) as well as operating frequencies and a long simulation time which has to be spent in order to reach the steady-state operation (after the transients characterized by large time constants due to system inertia have passed). A possible way to decrease simulation time is to change the abstraction level in which the system is defined, such that the system performs overall mostly in the same way but with less simulation complexity.

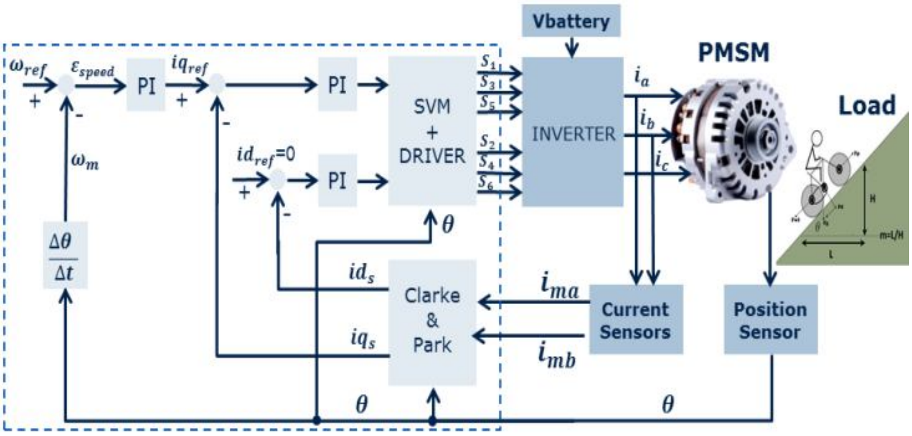

Fig. 1 – Diagram block of an E-Bike.

Given that some modeling languages like VHDL-AMS or Verilog-AMS reach rapidly their limits regarding simulation performance and interoperability, it is interesting to experiment such complex architectures with SystemC-AMS (http://www.systemc-ams.org.). The SystemC-AMS modeling language offers means to describe a system as a full virtual prototype, with software simulation capability, which can provide a thorough view of the system physical characteristics (electrical, mechanical etc.) that can be simulated with reduced license costs (Ting-Ting Liu *et al.*; Fan & Xuan, 2010; Lee, 2009; Dave, 2015; Jingyu *et al.*, 2011). It introduces two models of computation, SystemC-AMS synchronous dataflow (SDF) and SystemC-AMS conservative.

In addition to the SystemC event-driven model of computation (for digital description), we used such SystemC-AMS views to model the PMSM, the sensors used in the application and the Inverter Block.

An analytical method to set the saturation limits for the PI controllers is also presented and, in the last part simulation results obtained for the nominal case are given.

2. Application Description

Apart from the electric motor, the electrical part of an E-bike contains: a controller block, a power stage consisting of a transistor driver and an inverter circuit, sensors for electrical signals (three phase currents) and mechanical signals (motor position). All these components are presented in Fig.1. The PMSM motion is controlled by applying the Field-Oriented Control (FOC) algorithm (André Veltman) that has greatly improved the performance of AC motor drives. These control strategies use pulse width modulated (PWM) switching strategies aimed at producing a precisely controlled current in the windings of the motor.

All the components of the motor drive system were modeled in SystemC and SystemC AMS (SystemC-AMS, http://www.systemc-ams.org.):

a) The microcontroller was modeled as an algorithmic block, by using the Transaction Level Modeling (TLM) formalism.

b) The Driver block modeled with the Discrete Event (DE) formalism.

c) The Inverter circuit was modeled with the Discrete Event (DE) and Electrical Linear Networks (ELN) formalism.

d) Timed Data Flow (TDF) and ELN were used for the motor model.

e) ELN+ TDF were used for modeling the sensors.

The components are generic. Their behavior and internal structure is parameterized to a reasonable degree to allow their adaptation to different specifications.

3. Saturation Limits of PI Controllers According to the Current Sampling

System simulations done with both type of current sensors reveal that wrong saturation limits of the PI controllers can affect the electrical torque response in the startup phase. An improper saturation of the controllers will affect the reading process of the motor currents and the FOC algorithm will be corrupted. From the simulations it has been observed that special saturation must be done when Shunt resistor is used as a sensor for the currents.

According to the physical laws of inertia, the amount of torque necessary to start a load (starting torque, or Peak starting torque) is always much greater than the amount of torque required to maintain rotation of the load after it has achieved normal speed. In the case of a PMSM, the Peak starting torque is commonly limited to 140% of rated torque to avoid demagnetizing the field poles. If the control is correctly configured, after the startup (transient) phase, where the peak torque is constant and the mechanical speed is linearly increasing, the torque falls with the square of the voltage and current reduction. In order to maintain a constant output power, it is important that in this startup phase the peak torque is constant.

As a general practice, the values of the saturations for the current PI controllers are set to the maximum phase voltage applied to the motor windings, which cannot be higher than $V_{\text{Batt}}/\sqrt{3}$. This limit can be easily demonstrated. The three phase voltages applied on the motor windings are:

$$\begin{cases} V_a = A\cos\theta, \\ V_b = A\cos\left(\theta - \dfrac{2\pi}{3}\right), \\ V_c = A\cos\left(\theta + \dfrac{2\pi}{3}\right), \end{cases} \tag{1}$$

where: A is the amplitude of the AC voltages. For one “line to line” voltage (V_{ab}):

$$V_{ab} = V_a - V_b = A\left[\cos\theta - \cos\left(\theta - \frac{2\pi}{3}\right)\right] = -\sqrt{3}A\sin\left(\theta - \frac{\pi}{3}\right). \tag{2}$$

From the equation above we can find that the amplitude of V_{ab} voltage is: $\left|-\sqrt{3}A\right| = \sqrt{3}A$.

Therefore, the maximum amplitude of the line to line voltage is:

$$V_{ab\_\text{MAX}} = \sqrt{3}A = V_{\text{Batt}} \to A = \frac{V_{\text{Batt}}}{\sqrt{3}}. \tag{3}$$

For a three-phased motor, the stator voltage in *qd* frame is given by: $V_s = \sqrt{V_q^2 + V_d^2} = A$, which implies that the maximum value of V_s is:

$$V_{s\text{Max}} = \sqrt{V_{q\text{Max}}^2 + V_{d\text{Max}}^2} = \frac{V_{\text{Batt}}}{\sqrt{3}}. \tag{4}$$

Further simulations of this system has shown that these saturation limits are good enough only in the case wher the current signal (which has to be read) is a continuous one (for example when Hall sensors are used and placed between the outputs of the Inverter block and the motor terminals).

When the currents are sensed by using a Shunt Resistor, those currents are modulated (as a direct consequence of the SVM technique) and it is not feasible to read the currents at a random moment of time.

The SystemC model of the shunt resistor sensor is implemented using TDF (for gain stage) and the ELN approach (for the output RC filter). The voltage sensed by this sensor can be expressed as:

$$V_{\text{out}} = V_{\text{offset}} + \text{GAIN}\left(V_2 - V_1\right), \tag{5}$$

where: V_{offset} is the input offset of the amplifier; GAIN is the (open loop) gain of the amplifier; V_2 and V_1 represent the input terminal ("+" and "–") of the amplifier.

An output low pass filter is used in order to filter the high frequency spikes which appear, but its bandwidth is higher than the PWM frequency because the PWM signal should not be filtered. After all the signals on the shunt resistors have been filtered, amplification is required for each of them in order to adapt the signals to the range of voltage that can be read by the ADC peripheral embedded in the microcontroller unit.

The sample window available mainly depends on ADC sampling time, the ON time of low-side switches, dead time, and switching transient time. These will limit the high speed operation of the motor. We address here only the problem caused by the charging time of the low pass filter used in the output of the amplifier stage. There is a rise time issue associated with the RC time constants involved. This charging time is governed by the equation:

$$t = -\tau \ln\left(1 - \frac{V_c}{V_s}\right), \tag{6}$$

where: τ = RC, V_c is the voltage across the capacitor, V_s is the supply voltage t is the elapsed time since the application of the supply voltage and RC is the *time constant* of the RC charging circuit (0.484 μs). After an interval equivalent to 4 time constants, (4*T*) the capacitor of this RC charging circuit is virtually fully charged and the voltage across the capacitor is now about 99% of its maximum value, $0.99V_s$.

For V_c =99% V_s ↔0.99(2.5 V ± GAINR$_{\text{shunt}}I_{\text{sMAX}}$).

The obtained transient time is:

$$t = -\tau \ln\left(1 - 0.99\right) = 2.23\ \mu\text{s}. \tag{7}$$

This time delay is increased to3.5 [μs], on the one hand, because it must be a multiple of the time unit (500 [ns] - the PWM resolution) and on the other hand, because 1[us] is considered the dead time of the PWM signals.

Considering that the PWM resolution is set to 500 ns, the maximum ON time is:

a) $T_{\text{ON\_MAX}}$ = 50 μs – 3.5 us = 46.5 us – which corresponds to a duty cycle 46.5 μs/500 ns = 93% (the PWM counter will match the 93 value).

b) $T_{\text{ON\_MIN}}$ = 3.5 us – corresponds to a duty cycle 3.5 μs/500 ns = 7%.

Therefore, the current reading is done with a delay of 3.5 us after all low-side switches are ON, delay which provides the time for current to reach the stable value due to charging time. When the PWM counter matches the 93 value all the ADC data will be read. The time window for the ADC current is shown in Fig. 2. According to previous analysis, saturations of the PI controller are done in the microcontroller model. Therefore, the saturation of the Current PI controllers is:

$$V_{d\_\mathrm{SAT}},\ V_{q\_\mathrm{SAT}} = (\pm)93\% \frac{V_{\mathrm{Batt}}}{\sqrt{3}} = (\pm)19.33\ \mathrm{V}. \tag{8}$$

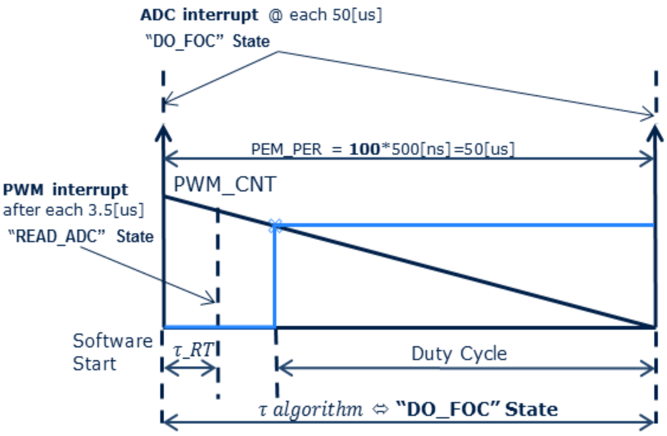

Fig. 2 – ADC reading time window.

The effect of PI controller saturations can be seen in Fig.3a (for low speed test) and Fig.4a (for high speed test) where saturation level ($V_s = \pm V_{\mathrm{Batt}} / \sqrt{3}$) is not enough in order to correctly read the stable value of the currents. In Figs. 3 *b* and 4 *b* the new saturation level ($V_s = 93\% \frac{V_{\mathrm{Batt}}}{\sqrt{3}}$) improves the currents waveforms by reading the true (stable) values of the motor currents. Therefore, the oscillations in torque and speed signals which can be seen in Figs. 3 *a* and 4 *a* are eliminated by this improvement done in microcontroller. In the same Fig. 4 the effect of the microcontroller reading process is illustrated with the signal denoted "dia". The effect of a wrong reading values of the motor currents has a direct impact on the FOC algorithm. Because the control is not working properlly, in transient phase where the currents loops are saturated (saturation needeed to develop the necessary Peak Torque) the electrical torque exceeds the Peak Torque value (56 Nm according to the motor specifications), while in the steady state the error from the currents is corrected by the negative feedback loops. This effect can be seen in Figs. 3 and 4. The effect of PI controller saturation on Peak Torque response can be seen in Fig. 5 *a* where the clamping is considered $V_s = \pm V_{\mathrm{Batt}} / \sqrt{3}$ and in Fig. 5 *b* where the clamping is considered $V_s = 93\% \frac{V_{\mathrm{Batt}}}{\sqrt{3}}$. In Figs. 6 *a* and 6 *b* "Vg1" is the gate voltage applied on the switch; "-Ia" is the motor current through phase "A", "dia" is the current reconstructed in the microcontroller, while "IaRs" is the current through shunt resistor which comes from phase "A" of the motor.

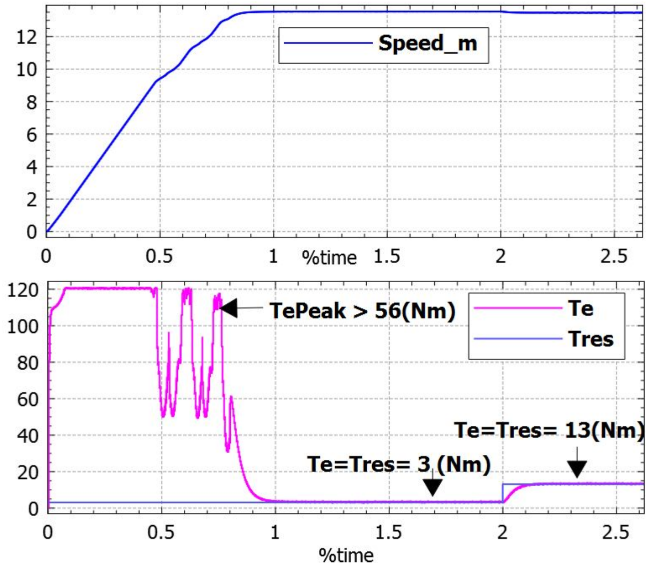

Fig. 3 *a* – The effect of PI controller saturation on Peak Torque response with $V_s = V_{\text{Batt}} / \sqrt{3}$ (low speed).

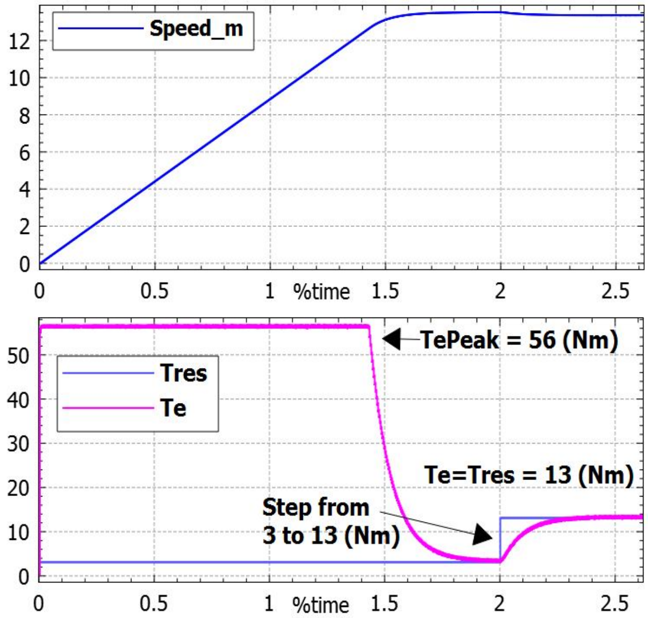

Fig. 3 *b* – The effect of PI controller saturation on Peak Torque response with $V_s = 93\% \frac{V_{\text{Batt}}}{\sqrt{3}}$ (low speed).

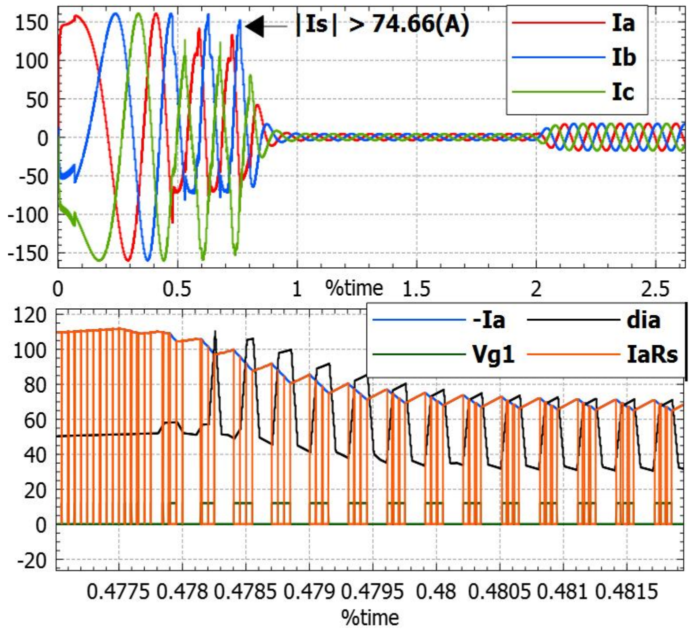

Fig. 4 *a* – The effect of PI controller saturation on motor ia, ib, ic currents (up) and the effect on currents sampling (down): $V_s = V_{\text{Batt}} / \sqrt{3}$ (low speed).

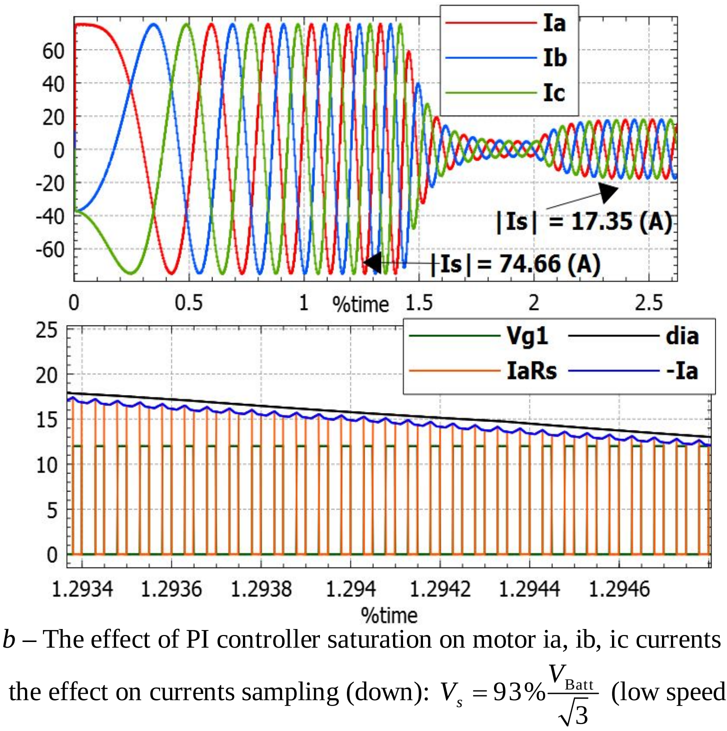

Fig. 4 *b* – The effect of PI controller saturation on motor ia, ib, ic currents (up) and the effect on currents sampling (down): $V_s = 93\% \frac{V_{\text{Batt}}}{\sqrt{3}}$ (low speed).

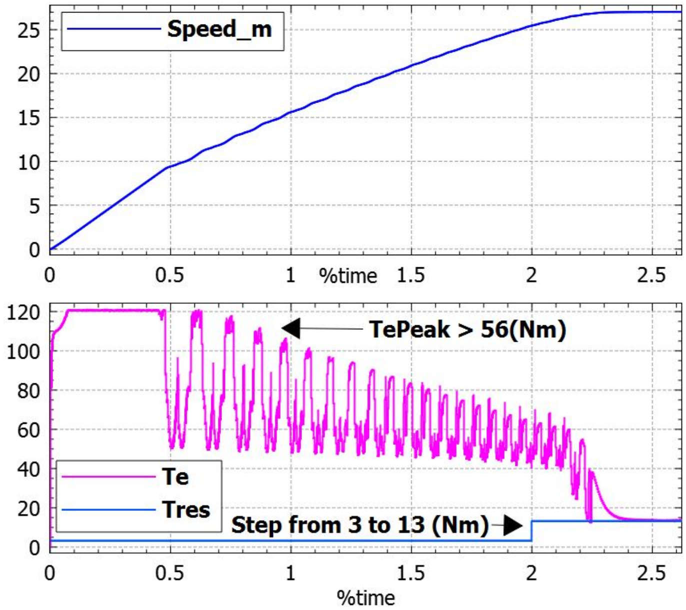

Fig. 5 *a* – The effect of PI controller saturation on Peak Torque response with $V_s = V_{\text{Batt}} / \sqrt{3}$ (high speed).

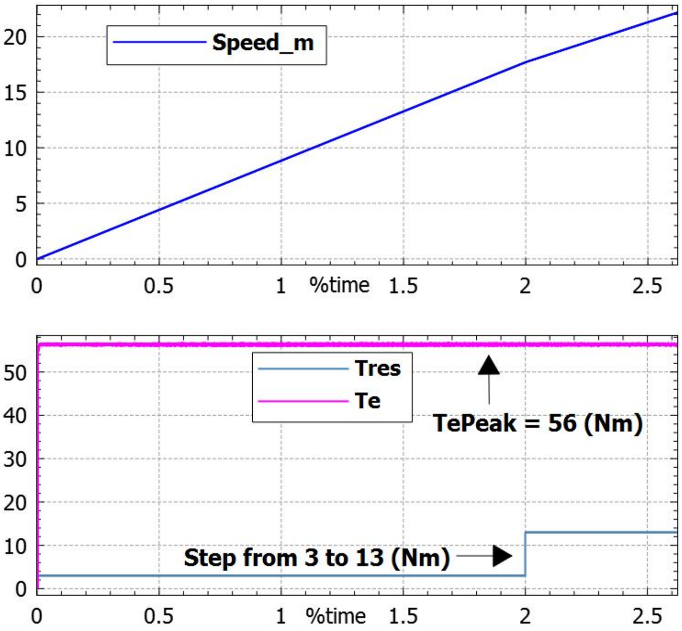

Fig. 5 *b* – The effect of PI controller saturation on Peak Torque response with $V_s = 93\% \frac{V_{\text{Batt}}}{\sqrt{3}}$ (high speed).

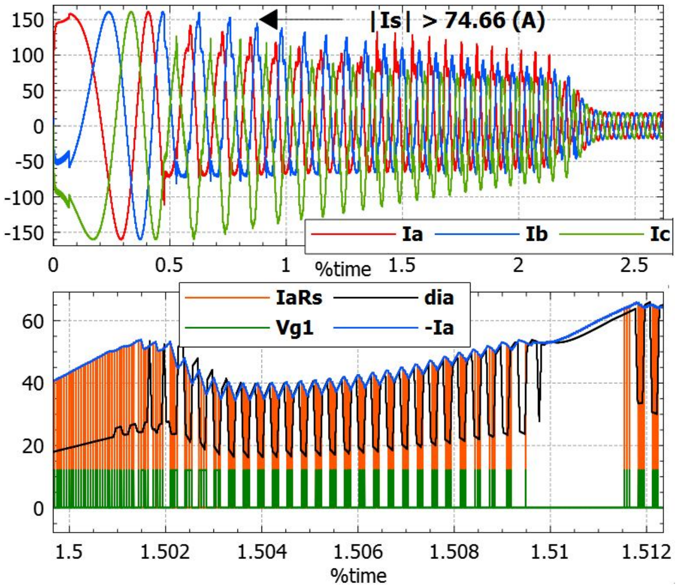

Fig. 6 *a* – The effect of PI controller saturation on motor ia, ib, ic currents (up) and the effect on currents sampling (down): $V_s = V_{\text{Batt}} / \sqrt{3}$ (high speed).

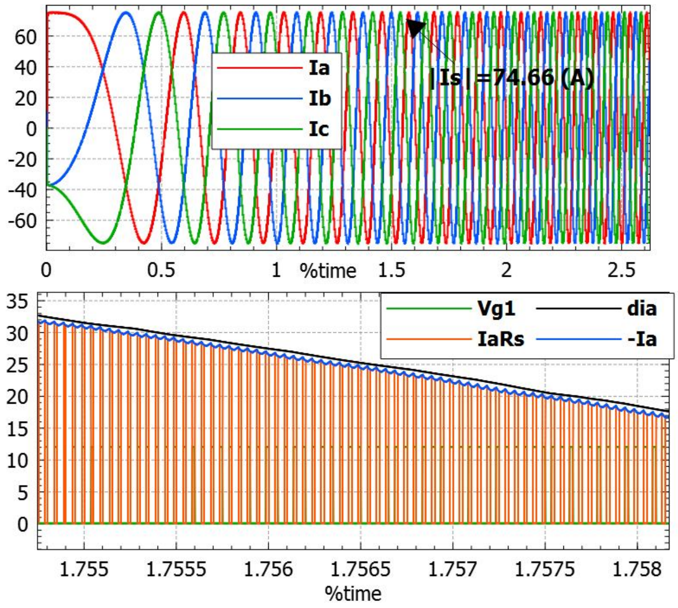

Fig. 6 *b* – The effect of PI controller saturation on motor ia, ib, ic currents (up) and the effect on current sampling (down): $V_s = 93\% \frac{V_{\text{Batt}}}{\sqrt{3}}$ (high speed).

5. Conclusions

In this paper we have presented an analog mixed signal application using SystemC and SystemC-AMS language. We show that, in order to achieve high system performances (such as low ripples in torque response), the control should be adjusted depending on the current sensor bandwidth. The adjustment is done by clamping the PI controllers to $93\%\frac{V_{\text{Batt}}}{\sqrt{3}}$. Also, we show that it is possible to model, validate and optimize a complex system using only one description language, SystemC-AMS (an open-source C++ library). The simulation of one run in SystemC-AMS takes between 5 to10 min compared to about 1h in VHDL-AMS. Moreover, we are able to simulate both digital and analog blocks simultaneously (because in Matlab/Simulink, for example, is more difficult to perform microcontroller models).

Acknowledgments. The current thesis is a project developed with the support of Infineon Technologies MUC, for which I am very thankful.

REFERENCES

Dave Wilson, *Teaching Your PI Controller to Behave*, Blogs: Motor Drive & Control, Motion Products Evangelist, Texas Instruments, 2015.

Fan Xuan, *Intelligent Power Assist Algorithms for Electric Bicycles*, Ph. D. Diss., University of California, Berkeley, CA, 2010; Available on: http://digitalassets.lib.berkeley.edu/etd/ucb/text/Fan_berkeley_0028E_11104.pdf)

Grellet G., Clerc G., *Actuators Electric Principe/Model/Control*, Eyrolles (2nd ed.), 2000, 298.

Grimm C., Barnasconi Vachoux A.M., Einwich K., *An Introduction to Modeling Embedded Analog/Mixed-Signal Systems using SystemC AMS extensions*, http://www.systemc.org/downloads/, 2008.

Hughes Austin, Bill Drury, *Electric Motors and Drives: Fundamentals, Types and Applications*, Newnes, 2013.

Jingyu Liu *et al*., *Steering (EPS) System Based on ADAMS*, Electric Information and Control Engineering (ICEICE), Internat. Conf., 2011.

Lee Stephens, *The Significance of Load to Motor Inertia Mismatch*, Senior Motion Control Engineer Kollmorgen, www.kollmorgen.com.

Pragasan P., Krishnan R., *Modeling of Permanent Magnet Motor Drives*, IEEE Trans. Industrial Electronics, **35**, *4* (1988).

Ting-Ting Liu *et al*., *Simulation of PMSM Vector Control System Based on Matlab/Simulink*, Measuring Technology and Mechatronics Automation, ICMTMA'09, **2**, IEEE, 2009.

Trzynadlowski A.M., *The Field Orientation Principle in Control of Induction Motors*, Kluwer Academic Publishers, 1994.

Veltman A., Duco W.J., Pulle D., de Doncker R., *Fundamentals of Electrical Drives*, Springer Netherlands, 2007.

Veltman A., Pulle D., de Doncker R., *Fundamentals of Electrical Drives*, Springer Netherlands, 2007.

* * * http://www.ebike.heinzmann.com/en/systems/directpower/motor.

* * * *SystemC-AMS*, http://www.systemc-ams.org.

PROIECTAREA LIMITĂRILOR ÎN REGULATOARELE PI PE BAZA EŞANTIONĂRII CURENȚILOR ÎN APLICAȚII DE CONTROLUL MOTOARELOR

(Rezumat)

Este prezentată o aplicație de semnal mixt analogic folosind limbajul de modelare SystemC şi SystemC-AMS. În lucrare se arată că, pentru a asigura performanțe bune ale sistemului (cum ar fi erorile de joasă frecventă ale cuplului electromecanic), controlul trebuie ajustat în funcție de lărgimea de bandă a senzorului de curent. Reglarea se face prin limitarea regulatoarelor PI, de pe buclele de curenți, la valoarea $93\%\frac{V_{\text{Batt}}}{\sqrt{3}}$. De asemenea, se arată că este posibilă modelarea, validarea şi optimizarea unui sistem complex, folosind doar un singur limbaj de descriere comportamentală, SystemC-AMS (bibliotecă C++ open-source). O simulare a sistemului în SystemC-AMS durează între 5 to10 min., comparativ cu circa 1 oră în VHDL-AMS. Mai mult decât atât, se poate modela şi simula simulatn atât blocuri digitale şi cât şi analogice (în Matlab / Simulink, de exemplu, este mult mai dificil de a realiza modele de microcontroler).