

APPLICATION OF A MODEL-BASED DESIGN TOOL X2C IN INDUCTION MACHINE VECTOR CONTROL

Nikola Lopac – Gordan Šegon – Neven Bulić*

Department of Automation and Electronics, Faculty of Engineering, University of Rijeka, Vukovarska 58, 51000 Rijeka, Croatia

ARTICLE INFO

Article history:

Received: 27.10.2017.

Received in revised form: 04.06.2018.

Accepted: 13.06.2018.

Keywords:

Model-based design

Embedded control systems

X2c tool

Digital signal processor

Induction machine

Vector control

DOI: <http://doi.org/10.30765/er.39.1.10>

Abstract:

Open source software tools for model-based design of embedded control systems represent a new method for rapid development and testing of advanced control structures in modern electric drives. The new concept of one of those tools, X2C, is presented in this paper, and in order to evaluate its performance in the development of complex control systems, it is applied to implement induction machine vector control. Based on the simulation model of the drive developed in PLECS, the parameters of the controllers were determined. The algorithm for the vector control of the induction machine was developed in X2C, implemented on the digital signal processor and applied to the real system. The system was tested during the no-load acceleration, deceleration, and reversing of the motor, while measuring the electrical and mechanical variables. Finally, the quantitative comparison of the experimental results and the results obtained by the equivalent simulations, based on the Integral Squared Error criterion, revealed that these results were well matched. This finding suggested that the control system was successfully implemented, thereby confirming the effectiveness of X2C tool in this particular type of application. With this work it has been shown that a transition from simulation to actual environment is rapidly achieved, with a simple verification of implemented methods accompanying the process. Realization of this work is a step forward in the utilization of open source software packages for implementation of induction machine vector control with the purpose of rapid verification of simulation models.

* Corresponding author. Tel.: +385 51 651413
E-mail address: neven.bulic@riteh.hr

1 Introduction

In recent years embedded systems have become widespread in many control systems applications [1]. As control algorithms become increasingly complex, the need for their simpler development and implementation on embedded systems using a model-based design is recognized [2]. This approach consists of the development of a mathematical block model of the entire system (including the control system and the controlled object itself), simulations prior to the implementation on the actual system, and automatic generation of the embedded code [3]. There are many reported benefits this approach brings to the development of the control algorithms, such as a shorter time required for the development, a rapid testing, simpler and faster modifications/upgrades of the system, cost savings, product quality improvement and the independence of the various embedded hardware architectures and the related programming languages [2, 4]. However, some of these claims are to a large extent based on the authors' personal opinions and anticipations, rather than on an empirical research. Furthermore, there are still some negative attitudes and a resistance to a wider application of the model-based design present among software engineers [5-7].

Several studies [8-14] dealt with the development of prototype frameworks and methodologies for the system modeling and the automatic code generation with the aim of application in the embedded systems. In addition, nowadays there are several commercial and established solutions for the model-based design, such as MATLAB Simulink. However, such tools are not suitable for the use in smaller companies due to their price. Therefore, the emphasis is put on the development of free solutions that can provide these companies and educational institutions all the benefits of model-based design in the form of a rapid prototyping, and thus a potential competitive advantage.

One of these free and open source tools is X2C [15], developed by Linz Center of Mechatronics GmbH (LCM). LCM's experience in using this software tool for the development of the controllers in the mechatronic systems is presented in [16], and the deeper insights are also given in [17], together with the application to the adaptive control of the cart and pendulum system. Moreover, in the [18] X2C was used for the development of a nonlinear

force and torque controller of a bearingless flux-switching drive.

In this study, we investigated the possibility of using a model-based design and X2C in the development of a complex and computationally demanding control algorithm. For that purpose, we used X2C tool to develop an induction machine vector control algorithm, which was then, in the combination with a digital signal processor (DSP), applied to the real laboratory system.

The rest of this paper is organized as follows: the second section explains the realized algorithm and the third section presents the corresponding simulation model; the fourth section gives an overview of the hardware and software employed during the experimental testing; in the fifth section the experimentally obtained results are compared to the results of the performed simulations; finally, the sixth section brings a conclusion.

2 Vector control

Vector control, also known as field oriented control (FOC), is an advanced algorithm that enables efficient and accurate control of alternating current (AC) machines in both steady state and various transient conditions [19-20]. It was named after its basic principle: AC machine control is achieved by controlling the magnitude and the phase angle of the current and voltage vectors [19]. Machine variables are transformed from the stationary 3-phase system to the rotating dq reference frame, thus providing decoupled control of the machine torque and flux by quadrature (q) and direct (d) stator current component respectively [19]. Therefore, control characteristics similar to those of direct current (DC) machines are possible [19].

Among many variants of vector control, the work presented in this paper focused on the direct rotor-flux-oriented control. Moreover, the selected control structure utilizes a speed information from an encoder mounted on the machine shaft (sensored control) and is also suitable for a machine supplied by a voltage source inverter. Finally, space vector modulation (SVM) [21] was chosen as a pulse width modulation (PWM) method for the control of six inverter's switches.

Theoretical considerations preceding the algorithm development, including definitions of reference frames, transformations, space phasors, machine parameters, stator and rotor voltage equations,

decoupling circuit (of an ideal drive) and flux model, were done according to [19].

The developed control structure was based on the standard approaches elaborated in [19-22] and its

principle is illustrated in Fig. 1. Equivalent structure was realized both in PLECS software [23] for the simulation purposes and in X2C for the control of the real system.

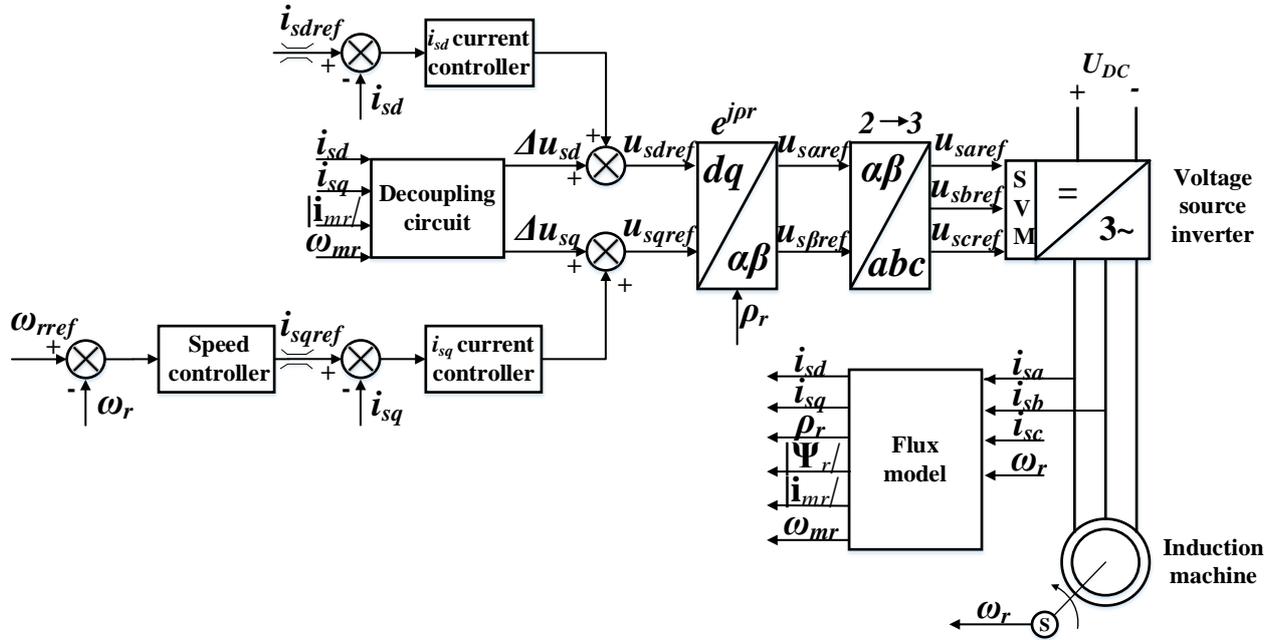


Figure 1. Schematic structure of sensored direct rotor-flux-oriented control for an induction machine supplied by a voltage source inverter.

Symbols used in Fig. 1 [19]:

- i_{sa}, i_{sb}, i_{sc} - 3-phase stator currents
- (u_{sa}, u_{sb}, u_{sc}) (voltages)
- i_{sd}, i_{sq} - direct and quadrature stator current (voltage) components expressed in the stationary ($\alpha\beta$) reference frame
- (u_{sd}, u_{sq}) - direct and quadrature stator current (voltage) components expressed in the rotor-flux-oriented (dq) reference frame
- ρ_r - phase angle between direct axes of the stator and the rotor-flux-oriented reference frame
- i_{mr} - space phasor of the rotor magnetizing current
- Ψ_r - space phasor of the rotor flux linkage
- ω_{mr} - angular speed of the rotor flux space phasor
- ω_r - angular rotor speed
- U_{DC} - DC link voltage

The control structure consisted of the outer speed and the inner current regulation loop. Three proportional-integral controllers were used: one speed controller and two current controllers (for d and q current component). The anti-windup method was also applied to these controllers. Proportional and integral gains of PI controllers with the parallel structure were set to the values in Table 1. These values were obtained using recommendations in [24], followed by the fine-tuning of the experimental controller.

Table 1. Parameters of PI controllers

Gain	Type	
	Speed	Current
Proportional	0.1084	8.1807
Integral	0.9453	12497.28

3 Simulation model

To support previous theoretical considerations, appropriate simulations were done in PLECS, using

$$J \frac{d\omega}{dt} = T_e - T_{load} - T_{loss} \quad (1)$$

where:

- J - moment of inertia (kgm²)
- ω - angular mechanical speed (rad/s)
- T_e - electromagnetic motor torque (Nm)
- T_{load} - load torque (Nm)
- T_{loss} - torque due to mechanical losses (Nm)

Table 3. Induction machine electrical parameters [26]

Parameter	Value
Stator resistance, R_s (Ω)	11.05
Stator leakage inductance, L_{sl} (H)	0.02
Rotor resistance, R_r (Ω)	6.11
Rotor leakage inductance, L_{rl} (H)	0.02
Magnetizing inductance, L_m (H)	0.29
Rotor inductance, L_r (H)	0.31
Number of pole pairs, p	2

In the steady state angular speed is assumed to be constant:

$$\frac{d\omega}{dt} = 0 \quad (2)$$

Moreover, motor operation at no-load was considered:

$$T_{load} = 0 \quad (3)$$

Therefore, in the steady state electromagnetic motor torque T_e was equal to the torque T_{loss} required to cover mechanical losses due to friction and windage:

$$T_e = T_{loss} \quad (4)$$

Developed electromagnetic torque is defined as [19]:

$$T_e = \frac{3}{2} p \frac{L_m^2}{L_r} |i_{mr}| i_{sq} \quad (5)$$

A torque was modelled here by the second degree polynomial due to mechanical losses T_{loss} :

$$T_{loss} = T_e = k_2 \omega^2 + k_1 \omega + k_0 \quad (6)$$

where quadratic term $k_2 \omega^2$ represents torque due to windage losses, linear term $k_1 \omega$ represents the viscous friction and constant term k_0 represents the idealized Coulomb friction.

Model coefficients k_2 , k_1 , k_0 were estimated using least-squares method to fit data. For that purpose, experimental measurements were conducted during the steady state motor operation. Angular speed was changed with a step of 10% of the nominal value and electromagnetic torque was calculated according to (5), based on the measured values of currents i_{mr} and i_{sq} . The measurement results are summarized in Table 4.

Table 4. Angular speed and electromagnetic torque measurements

Angular speed ω (rad/s)	Torque T_e (Nm)
18.064	0.00425
36.128	0.00900
54.192	0.01264
72.256	0.01548
90.320	0.01769
108.384	0.01979
126.448	0.02140
144.512	0.02304
162.576	0.02478
180.640	0.02586

Coefficient estimates were calculated applying the least-squares method that minimizes the sum of squared residuals (errors). For the i -th data point residual r_i is defined as:

$$r_i = T_i - \hat{T}_i \quad (7)$$

where:

- T_i - measured value of the electromagnetic torque (observed response value)
- \hat{T}_i - fitted value of the electromagnetic torque (predicted response value)

The sum of squared residuals S for n data points is defined as:

$$S = \sum_{i=1}^n r_i^2 = \sum_{i=1}^n (T_i - \hat{T}_i)^2 \quad (8)$$

where fitted torque value is defined as:

$$\hat{T}_i = k_2 \omega_i^2 + k_1 \omega_i + k_0 \quad (9)$$

To determine the coefficient estimates by minimizing the sum of squared residuals, the sum S is differentiated with respect to each coefficient and result is set to equal 0:

$$\frac{\partial S}{\partial k_2} = 0, \frac{\partial S}{\partial k_1} = 0, \frac{\partial S}{\partial k_0} = 0 \quad (10)$$

The normal equations in matrix form are obtained:

$$(\mathbf{W}^T \mathbf{W}) \mathbf{k} = \mathbf{W}^T \mathbf{T} \quad (11)$$

where:

\mathbf{W} is 10x3 design matrix defined as:

$$\mathbf{W} = \begin{bmatrix} \omega_1^2 & \omega_1 & 1 \\ \omega_2^2 & \omega_2 & 1 \\ \vdots & \vdots & \vdots \\ \omega_{10}^2 & \omega_{10} & 1 \end{bmatrix} \quad (12)$$

with \mathbf{W}^T denoting its transpose.

\mathbf{T} is 10x1 vector of measured torque values defined as:

$$\mathbf{T} = \begin{bmatrix} T_1 \\ T_2 \\ \vdots \\ T_{10} \end{bmatrix} \quad (13)$$

\mathbf{k} is 3x1 vector of unknown coefficients defined as:

$$\mathbf{k} = \begin{bmatrix} k_2 \\ k_1 \\ k_0 \end{bmatrix} \quad (14)$$

The least squares solution which estimates vector of coefficients is determined as:

$$\mathbf{k} = (\mathbf{W}^T \mathbf{W})^{-1} \mathbf{W}^T \mathbf{T} \quad (15)$$

Finally, following values of coefficient estimates are obtained:

$$\mathbf{k} = \begin{bmatrix} -6.000 \cdot 10^{-7} \\ 2.372 \cdot 10^{-4} \\ 8.310 \cdot 10^{-4} \end{bmatrix} \quad (16)$$

The fitted values of torque are calculated as:

$$\hat{\mathbf{T}} = \mathbf{W} \mathbf{k} \quad (17)$$

Fig. 4 depicts measured data points and fitted curve obtained for coefficient values determined by the least-squares method and given in (16). As seen from Fig. 4, the fitted curve is appropriate approximation of the measured data.

Moment of inertia J was determined in the second experiment by motor retardation test. Motor with unconnected load was running at initial speed ω_0 equal to nominal when the power supply was switched off. Motor gradually slowed down and stopped within 9.504 s. During this time interval motor was decelerated by the loss torque. Motor speed was measured as a function of time $\omega(t)$, which is depicted graphically in Fig. 5.

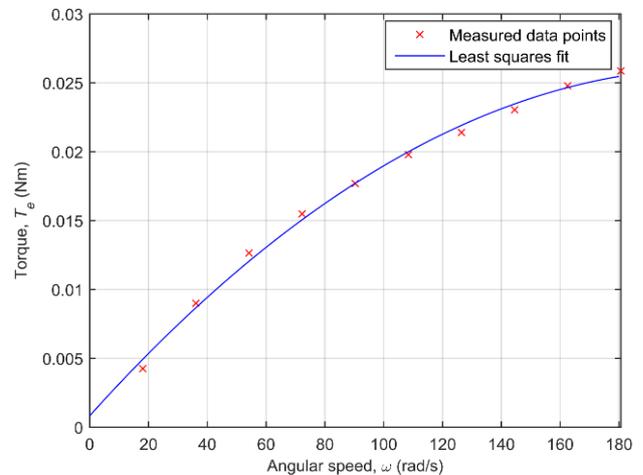


Figure 4. Measured data points and curve fitted by least-squares method.

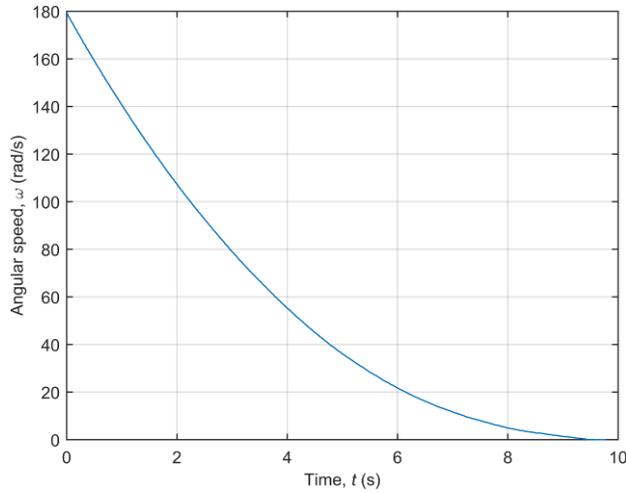


Figure 5. Motor angular speed during the retardation test.

During the time interval 0-1 s speed characteristic was linear. Hence, this interval was considered to determine moment of inertia. From equation of motion given by (1), it follows:

$$J \frac{d\omega}{dt} = -k_1 \bar{\omega} \quad (18)$$

where $\bar{\omega}$ is average angular speed during this interval (0-1 s).

Therefore, moment of inertia may be determined as:

$$J = \frac{-k_1 \bar{\omega}}{\frac{d\omega}{dt}} \quad (19)$$

Average speed $\bar{\omega}$ was calculated as:

$$\bar{\omega} = \frac{\omega(0s) + \omega(1s)}{2} = \frac{180 + 141.68}{2} \quad (20)$$

$$\bar{\omega} = 160.84 \text{ rad/s} \quad (21)$$

Term $d\omega/dt$ was obtained as a curve slope in the initial point ω_0 :

$$\frac{d\omega}{dt}(\omega_0) = \frac{\Delta\omega}{\Delta t} = \frac{141.68 - 180}{1} \quad (22)$$

$$\frac{d\omega}{dt}(\omega_0) = -38.32 \text{ rad/s} \quad (23)$$

Finally, moment of inertia was obtained:

$$J = 9.828 \cdot 10^{-4} \text{ kgm}^2 \quad (24)$$

Estimated mechanical parameters are finally summarized in Table 5.

Table 5. Induction machine mechanical parameters

Moment of inertia, J (kgm ²)	$9.828 \cdot 10^{-4}$
Quadratic term coefficient, k_2 (Nms ²)	$-6.000 \cdot 10^{-7}$
Linear term coefficient, k_1 (Nms)	$2.372 \cdot 10^{-4}$
Constant term, k_0 (Nm)	$8.310 \cdot 10^{-4}$

"The High Voltage Digital Motor Control (DMC) and Power Factor Correction (PFC) kit" [27], produced by Texas Instruments, was used to supply the motor with a variable 3-phase voltage.

This laboratory kit includes a 3-phase inverter with the maximum DC link voltage of 350 VDC [27]. The inverter has three pairs of IGBT switches and can be used to control 3-phase machines with the maximum output power of up to 1.5 kW [27]. Besides induction machines, it can also be used to control permanent magnets synchronous machines and brushless DC machines [27].

The control board contains fast QEP and CAP input pins suitable for speed and angle measurements [27]. In this concrete example digital signals from an 11-bit incremental optical encoder mounted on the machine shaft were connected to the four 32-bit QEP pins.

The board also features high-precision analog-to-digital converters (ADCs) and high-speed operational amplifiers for current measurements [27]. Two phase currents of the machine were measured using shunt resistors located in the two branches of the inverter, below the lower switch in the branch. The inverter's total return current and the DC link voltage were also measured.

These measurements represented voltage measurements, which were sampled by the "Sample & Hold" (S&H) units of the ADC and converted from analog values to digital values stored in the registers of the appropriate ADC channels afterwards. This 12-bit ADC has 16 channels and two S&H units, the ability to simultaneously convert two channels and the conversion speed of 12.5 MSPS (80 ns) [28].

The sampling took place once during the switching period, i.e. the PWM period, which was 100 μ s. The sampling needed to be synchronized with the PWM signal in a way that it was carried out every time the

three lower switches were switched on. This sampling mode ensured that the current passed through the shunts at the time the sampling was executed and thus the real current value was captured. Otherwise, S&H units would sample only the noise value that existed at the time the current did not pass through the switches. The analysis of the generated PWM signal showed that, due to its symmetry, the sampling was best carried out in the middle of the PWM period, since then all three upper switches were switched off, i.e. all three lower switches were switched on. In fact, to ensure that all current transients during the switching were finished, the sampling was done 2.67 μs after the middle point of the PWM period.

This laboratory kit supports an isolated JTAG emulation that enabled connection between the development software on the PC and the DSP [27]. USB ports were used for that purpose. There are several outputs available on the board to connect an oscilloscope in order to measure the signals from six PWM channels or to observe the phase voltages waveforms [27]. These features were useful during the development process and the control system analysis. The control software was implemented on the TMS320F28335 Delfino DSP [28] that belongs to the F2833x family of processors, which is a part of the Texas Instruments C2000 platform. In addition to the 32-bit fixed-point architecture, which is standard for Texas Instruments DSPs, these processors also have a 32-bit floating-point unit [28]. This unit allows the execution of mathematical operations and other manipulations over the numerical values written using the floating-point representation [28]. This project was realized utilizing floating-point unit.

TMS320F28335 is able to work with a frequency up to 150 MHz and supports programming in C/C++ and the assembler [28]. This processor can use watchdog timer and three 32-bit CPU timers [28]. There are 88 general purpose pins (GPIOs) that can be defined to be used as input or output pins [28]. Specific purposes may also be assigned to the GPIOs by configuring the appropriate registers [28].

4.2 Software setup

The vector control algorithm was realized in X2C tool using the basic template developed by Linz Center of Mechatronics as a basis.

Additionally, DSP hardware was configured using Code Composer Studio – an integrated development environment (IDE) supporting Texas Instruments processors. This IDE was utilized for the modifications of the C-code in the template, including configuration of the ADC sampling time, adjustment of the QEP registers, modifications of the PWM parameters and definitions of the input and output pins. Code Composer Studio was also used to compile generated C-code before downloading it to the target processor.

X2C is a software tool that enables the development of control structures for various systems based on the graphical programming and the system modeling, using programming blocks similar to those of MATLAB Simulink [15-17]. Developed control algorithm can be simulated before the deployment on the processor [15-17]. X2C allows the C-code to be automatically generated from a graphical model [15-17]. This code is used by the DSP and is suitable to be analyzed and edited by the developer [15-17]. X2C supports various microprocessor models from different manufacturers [15-17]. Therefore, developed control software is independent of the target processor's architecture [15-17].

In addition, X2C provides real-time operation, which includes an online visualization of the system variables using graphical user interface (GUI) *Scope* [15-17]. *Scope* offers functions that are found in many standard oscilloscopes and plotting tools [15-17]. These functions include modifying the graphical display, monitoring and recording data of interest [15-17].

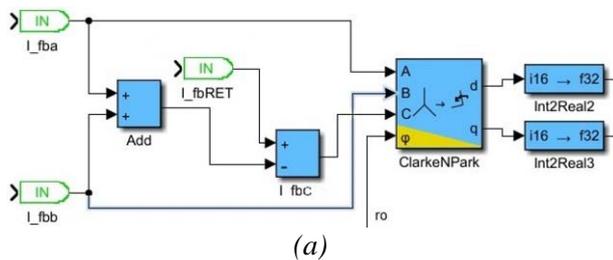
X2C also enables online modification of the block parameters, using the interface *Communicator* [15-17]. In this way, the behavior of the control system can be adjusted in real time, with automatic changes in the DSP code [15-17]. *Communicator* establishes connection between X2C and DSP [15-17]. It is also used to create C-code from X2C model and to download it to the DSP [15-17].

In addition to these features, X2C is free and based on open source platform for numerical computations Scilab/Xcos [15, 17, 29]. X2C may also be used with MATLAB Simulink environment [15, 17], as it was in this study.

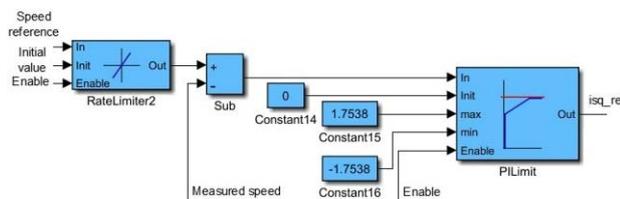
X2C programming blocks are available in several libraries, including *Control*, *General*, *Math* and *Motor control* [15]. These blocks are predefined, and no modification was made on them for purpose

of this study, but existing blocks were combined to build a complete control system. Detailed description of the blocks is available in [30]. In this project, floating-point blocks were used.

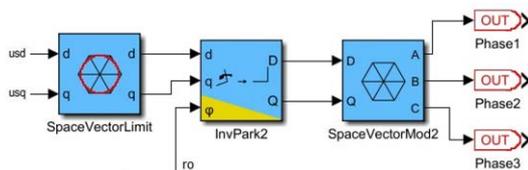
The developed graphical control structure is illustrated by three examples in Fig. 6. Part of the structure responsible for the transformation of the measured currents is shown in Fig. 6 (a). Three input blocks *IN* carried the digital per-unit current values with fixed-point representation. The third phase current was obtained by subtracting two measured phase currents from inverter's return current, using blocks *Add* and *Sub*. Phase currents, together with the estimated phase angle of the rotor flux, entered the block *ClarkeNPark* that performed Clarke's and Park's current transformations into the *dq* reference frame. The obtained values were further converted to values in amperes (floating-point representation) using blocks *Int2Real*, which multiply them with the maximum current that could be obtained by this type of measurement (16.5 A).



(a)



(b)



(c)

Figure 6. X2C control structure (a) Current transformation (b) Speed PI controller (c) Space vector modulation.

Fig 6 (b) shows the speed PI controller realized in X2C, with the corresponding input and output

variables. The ramp function was achieved by passing the speed reference through the block *RateLimiter*. Main input to the controller block *PILimit* was the difference between the speed reference and the measured speed obtained by differentiating the rotor angle signal from the encoder. Other inputs of the controller were used to set initial, minimum, and maximum output values.

The space vector modulation subsystem is presented in Fig. 6 (c). This subsystem consisted of three blocks: *SpaceVectorLimit*, *InvPark* and *SpaceVectorMod*.

SpaceVectorLimit block was used to limit the maximum magnitude of the stator voltage space phasor. *InvPark* block performed the inverse Park transformation of the voltages from the *dq* to the *αβ* reference frame. The last block, *SpaceVectorMod*, was responsible for generating three control signals for the inverter's switches – the results of the SVM algorithm execution. Those signals represented duty cycles (value range 0-1) for the upper switches of the inverter. The signals for the lower switches were obtained by inverting those for the upper switches, with the dead time of 3.6 μs.

4.3 Testing procedure

In order to evaluate the quality of the developed control system, three experiments were conducted to test the behavior of the drive under the dynamic conditions of motor free (no-load) operation. The nominal line voltage was applied to the machine. Electrical and mechanical variables were measured using sensors and were observed in X2C's *Scope*. The identical procedures were conducted in the equivalent PLECS simulations.

During the first experiment, motor was accelerated from the stall to the nominal speed. The speed reference in X2C software was gradually increased from 0 rpm to 1725 rpm, following the ramp function during the time interval of 0.488 s.

During the second experiment, motor was decelerated from the nominal to the zero speed. The speed reference in X2C was decreased from 1725 rpm to 0 rpm, following the ramp function with the negative slope during the time interval of 0.488 s.

The third experiment included motor reversing. The speed reference in X2C was first increased by ramp function from 0 rpm to the nominal value of 1725 rpm in a time of 0.488 s. After that, the speed was kept constant and equal to the nominal for 3 s. Then

the reversing started, i.e. the speed was decreased from the nominal value to 0 rpm, and then further to the negative nominal value, following the ramp function with the negative slope and total duration of 0.976 s. The negative speed represented motor rotation in the opposite direction. The speed was then kept at negative nominal value for 3 s and later increased again to the positive nominal value.

5 The results and discussion

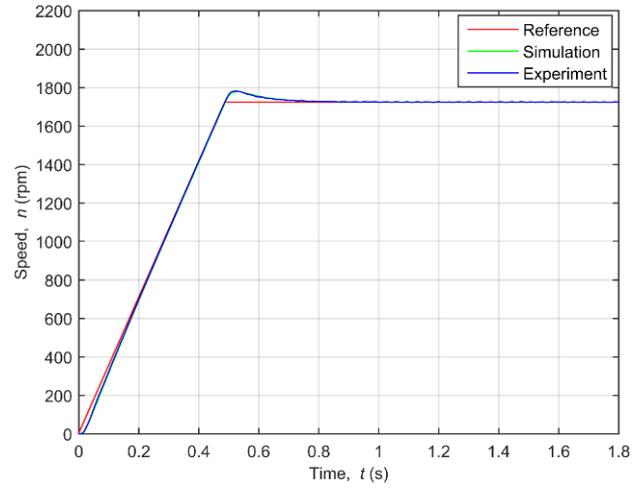
Simulation results and experimental measurements obtained during the previously described experiments are graphically presented in the following subsections. Moreover, simulation and experimental results are compared using quantitative measure in the form of Integral Squared Error (ISE) criterion. This measure of control system performance was calculated for simulated and experimentally measured values of speed, direct stator current component, and quadrature stator current component, with respect to their reference values. The ISE criterion is defined as:

$$ISE = \int_0^{\infty} e^2(t)dt \quad (25)$$

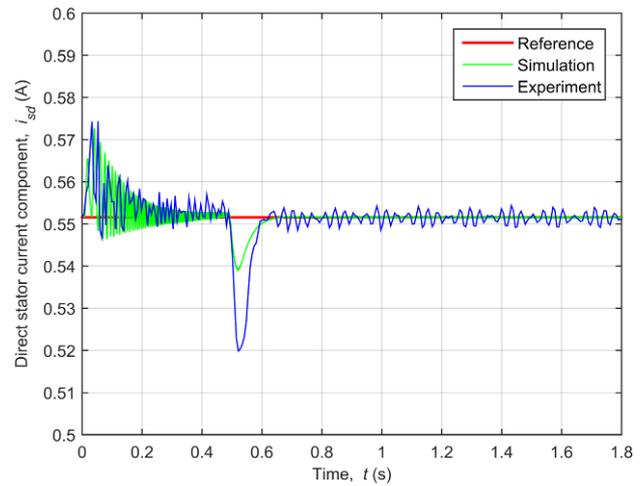
where $e(t)$ is the error, i.e. the difference between simulated/experimental values and their respective reference values.

5.1 Motor acceleration

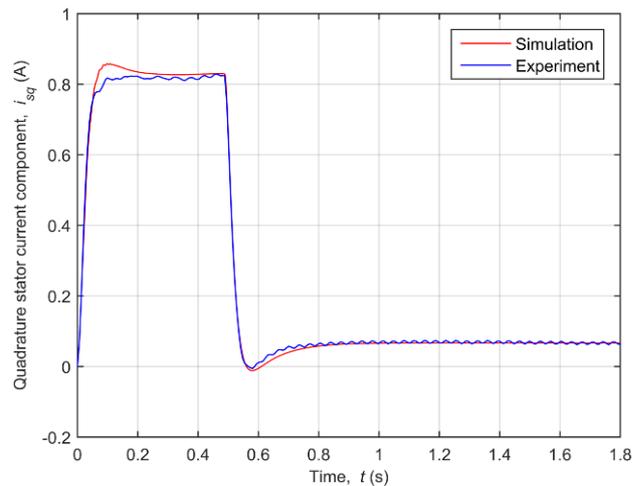
Simulated and experimentally measured variables obtained during motor acceleration from stall are depicted in Fig. 7. As seen from Fig. 7, experimentally obtained values, including speed shown in Fig. 7 (a), direct stator current component shown in Fig. 7 (b), reference of quadrature stator current component shown in Fig. 7 (c) and



(a)



(b)



(c)

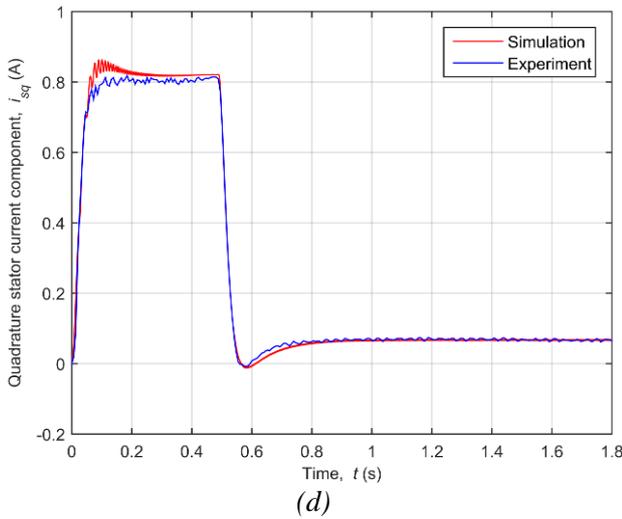


Figure 7. Simulation and experimental results during no-load acceleration of the induction motor (a) Speed (b) Direct stator current component (c) Reference of quadrature stator current component (d) Quadrature stator current component.

quadrature stator current component shown in Fig. 7 (d), are well matched with the simulated values. This is confirmed by quantitative measures, summarized in Table 6 and Table 7.

The speed response during motor acceleration was analyzed and comparison between performance measures of simulated and experimentally obtained response is shown in Table 6. This comparison indicates that simulated and experimental response exhibit close values of maximum overshoot, peak time and settling time. Moreover, these values in general indicate good dynamic characteristics of the developed control system.

Table 6. Performance measures of speed response

	Simulation	Experiment
Maximum overshoot, M_p (%)	3.14	3.36
Peak time, t_p (s)	0.5295	0.5287
Settling time (2%), t_s (s)	0.5900	0.5803

Table 7 provides values of the ISE criteria computed for speed, direct and quadrature stator current component during motor acceleration. Close values of the ISE criteria obtained for simulated and

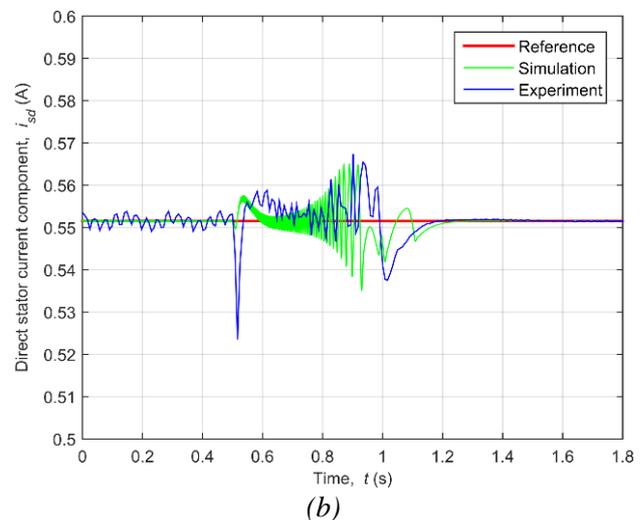
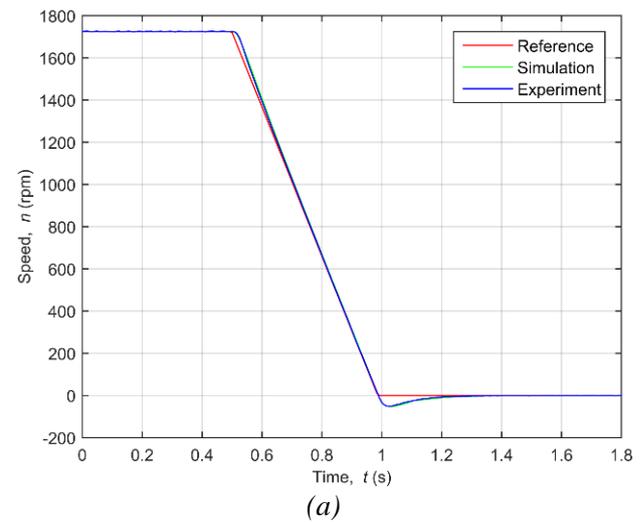
experimental responses indicate that simulation and experimental results are well matched.

Table 7. The ISE criteria for motor variables during acceleration

	ISE	
	Simulation	Experiment
Speed n	$5.2767 \cdot 10^2$	$5.3233 \cdot 10^2$
Direct stator current component i_{sd}	$1.6169 \cdot 10^{-5}$	$5.9533 \cdot 10^{-5}$
Quadrature stator current component i_{sq}	$1.9301 \cdot 10^{-4}$	$4.7905 \cdot 10^{-4}$

5.2 Motor deceleration

Fig. 8 depicts simulated and experimental measurements obtained during motor deceleration from the nominal speed.



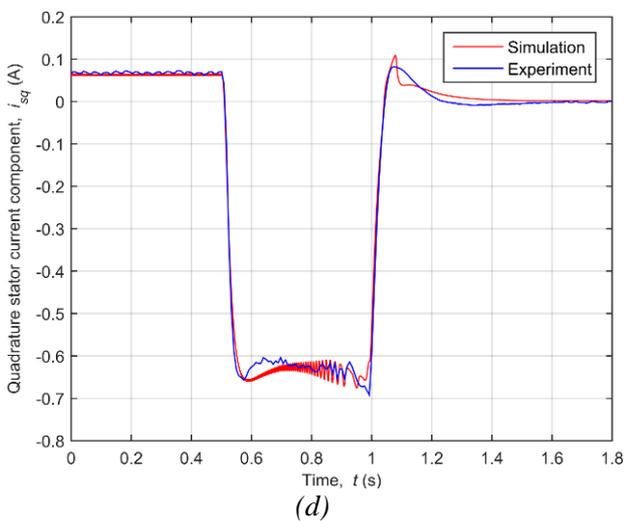
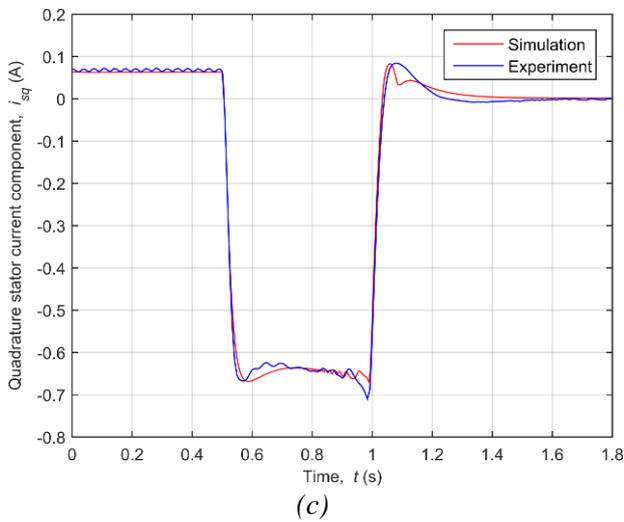


Figure 8. Simulation and experimental results during no-load deceleration of the induction motor (a) Speed (b) Direct stator current component (c) Reference of quadrature stator current component (d) Quadrature stator current component.

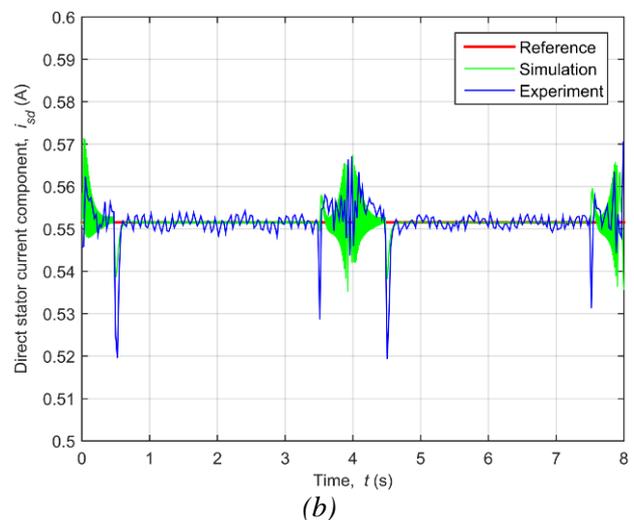
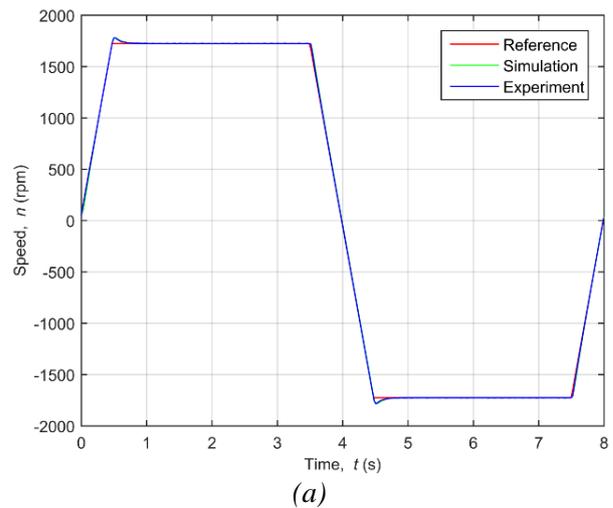
Graphical representations of speed (Fig. 8 (a)), direct stator current component (Fig. 8 (b)), reference of quadrature stator current component (Fig. 8 (c)) and quadrature stator current component (Fig. 8 (d)) show that simulation and experimental values are well matched. This is additionally confirmed by similar values of the ISE criteria obtained for simulated and experimental values of speed, direct and quadrature stator current component, which are given in Table 8.

Table 8. The ISE criteria for motor variables during deceleration

	ISE	
	Simulation	Experiment
Speed n	$5.0753 \cdot 10^2$	$4.3674 \cdot 10^2$
Direct stator current component i_{sd}	$1.3259 \cdot 10^{-5}$	$3.0710 \cdot 10^{-5}$
Quadrature stator current component i_{sq}	$3.9873 \cdot 10^{-4}$	$4.2217 \cdot 10^{-4}$

5.3 Motor reversing

Simulated and experimental values obtained during the third experiment (motor reversing) are depicted in Fig. 9, including speed (Fig. 9 (a)), direct stator current component (Fig. 9 (b)), reference of quadrature stator current component (Fig. 9 (c)), and quadrature stator current component (Fig. 9 (d)).



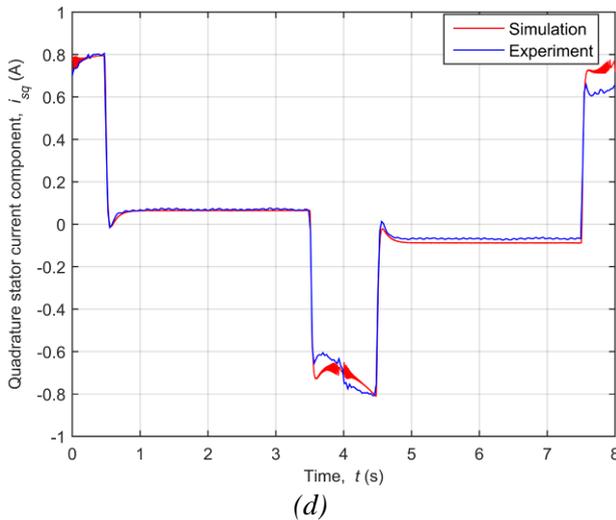
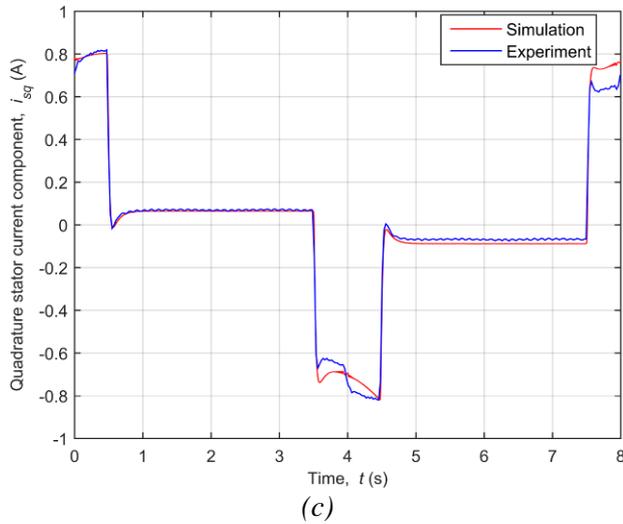


Figure 9. Simulation and experimental results during no-load reversing of the induction motor (a) Speed (b) Direct stator current component (c) Reference of quadrature stator current component (d) Quadrature stator current component.

These graphical representations indicate again that simulation and experimental results are well matched. Close values of the ISE criteria calculated for simulated and experimental values of speed, direct and quadrature stator current component confirm this. Calculated values of the ISE criteria are given in Table 9.

Table 9. The ISE criteria for motor variables during reversing

	ISE	
	Simulation	Experiment
Speed n	$1.5107 \cdot 10^3$	$1.0036 \cdot 10^3$
Direct stator current component i_{sd}	$5.0184 \cdot 10^{-5}$	$1.7805 \cdot 10^{-4}$
Quadrature stator current component i_{sq}	$0.8390 \cdot 10^{-3}$	$1.3178 \cdot 10^{-3}$

The above conducted comparison of simulated and experimentally obtained responses, based on the ISE criterion as a quantitative measure, indicated that simulated and experimental results were well matched during the dynamic conditions of three experiments: motor acceleration, deceleration, and reversing. These observations suggest that the control system, employing vector control of small laboratory induction motor, has been successfully realized by applying X2C tool.

6 Conclusion

The successful implementation of the described control system proves the possibility of X2C software to be applied as a tool for verification of simulation models of laboratory induction machines vector control systems. Being a free software, it may be an appropriate solution for rapid prototyping of advanced vector control systems in small companies. The X2C motor control template was created as a product of this work, requiring minimal knowledge of C programming language and microprocessor architectures. Therefore, it may be used in electrical drives education purposes, where various vector control schemes for laboratory induction machines can be achieved by simple parameters changing, enabling students both simulations and applications to the actual systems.

References

- [1] Isermann, R.: *Mechatronic systems—Innovative products with embedded control*, Control Engineering Practice, 16 (2008), 1, 14–29.
- [2] Selic, B.: *The pragmatics of model-driven development*, IEEE Software, 20 (2003), 5, 19–25.

- [3] France, R., Rumpe, B.: *Model-driven Development of Complex Software: A Research Roadmap*, Conference on Future of Software Engineering, Minneapolis, USA, 2007, 37-54.
- [4] Broy, M., Kirstan, S., Krcmar, H., Schätz, B., Zimmermann, J.: *What is the benefit of a model-based design of embedded software systems in the car industry?*, in *Emerging Technologies for the Evolution and Maintenance of Software Models*, IGI Global, Hershey, 2011, 343-369.
- [5] Agner, LTW., Soares, IW., Stadzisz, PC., Simao, JM.: *A Brazilian survey on UML and model-driven practices for embedded software development*, Journal of systems and software, 86 (2013), 4, 997-1005.
- [6] France, R., Rumpe, B.: *Problems and opportunities for model-centric versus code-centric software development: a survey of software professionals*, Proceedings of the 2008 international workshop on Models in software engineering, Leipzig, Germany, 2008, 27-32.
- [7] Hutchinson, J., Whittle, J., Rouncefield, M.: *Empirical Assessment of MDE in Industry*, 33rd International Conference on Software Engineering, Honolulu, USA, 2011, 471-480.
- [8] Rizano, T., Passerone, R., Macii, D., Palopoli, L.: *Model-based design of embedded control software for hybrid vehicles*, 6th IEEE International Symposium on Industrial Embedded Systems, Vasteras, Sweden, 2011, 75-78.
- [9] Chehade, WE., Radermacher, A., Terrier, F., Selic, B., Gerard, S.: *A Model-Driven Framework for the Development of Portable Real-time Embedded Systems*, 16th IEEE International Conference on Engineering of Complex Computer Systems, Las Vegas, USA, 2011, 45-54.
- [10] Wehrmeister, MA., Pereira, CE., Rammig, FJ.: *Aspect-Oriented Model-Driven Engineering for Embedded Systems Applied to Automation Systems*, IEEE transactions on industrial informatics, 9 (2013), 4, 2373-2386.
- [11] Wehrmeister, MA., Freitas, EP., Binotto, APD., Pereira, CE.: *Combining aspects and object-orientation in model-driven engineering for distributed industrial mechatronics systems*, Mechatronics, 24 (2014), 7, 844-865.
- [12] Moallemi, M., Wainer, G.: *Modeling and simulation-driven development of embedded real-time systems*, Simulation Modelling Practice and Theory, 38 (2013), 115-131.
- [13] Doukas, G., Thramboulidis, K.: *Real-Time-Linux-Based Framework for Model-Driven Engineering in Control and Automation*, IEEE transactions on industrial electronics, 58 (2011), 3, 914-924.
- [14] Ma, L., Xia, F., Peng, Z.: *Integrated design and implementation of embedded control systems with Scilab*, Sensors, 8 (2008), 9, 5501-5515.
- [15] Linz Center of Mechatronics GmbH: *About X2C* [Online], Available: <http://www.mechatronic-simulation.org/x2c/x2c-home/> [Accessed: 15-Oct-2017].
- [16] Hochwallner, M., Fragner, S.: *Experience in Model Based Design with the Tool X2C in Controller Design in a Mechatronic Product Development Environment*, Presentation, 8th MODPROD Workshop on Model-Based Product Development, Linköping, Sweden, 2014.
- [17] Grabmair, G., Mayr, S., Hochwallner, M., Aigner, M.: *Model based control design – a free tool-chain*, European Control Conference, Strasbourg, France, 2014, 826-831.
- [18] Radman, K., Gruber, W., Bulić, N.: *Control design of a bearingless flux-switching slice drive*, Proceedings of the 6th European Embedded Design in Education and Research Conference, Milano, Italy, 2014, 197-201.
- [19] Vas, P.: *Sensorless vector and direct torque control*, Oxford University Press, Oxford, 1998.
- [20] Bose, BK.: *Modern Power Electronics and AC Drives*, Prentice Hall, Upper Saddle River, 2002.
- [21] Quang, NP., Dittrich, JA.: *Vector Control of Three-Phase AC Machines*, Springer, Berlin, 2008.
- [22] Leonhard, W.: *Control of Electrical Drives*, Springer, Berlin, 2001.
- [23] Plexim GmbH: *PLECS User Manual Version 4.0*, Zürich, 2016.
- [24] Wilson, D.: *Teaching Your PI Controller to Behave* [Online], Available: https://e2e.ti.com/blogs_/b/motordrivecontrol/a

- rchive/2015/07/20/teaching-your-pi-controller-to-behave-part-i [Accessed: 18- Oct- 2017].
- [25] Grainger: *1/4 HP General Purpose Motor* [Online], Available: <https://www.grainger.com/product/MARATHON-MOTORS-1-4-HP-General-Purpose-Motor-4N579> [Accessed: 18- Oct- 2017].
- [26] Texas Instruments: *ControlSUITE: HVMotorCtrlPfcKit_v1.7, HVACI_Sensored-Settings.h* [CD-ROM], 2010.
- [27] Texas Instruments: *High Voltage Motor Control and PFC Kit Hardware Reference Guide*, Dallas, 2010.
- [28] Texas Instruments: *TMS320F2833x, TMS320F2823x Digital Signal Controllers (DSCs)*, Dallas, 2016.
- [29] Scilab Enterprises: *About Scilab* [Online], Available: <https://www.scilab.org/scilab/about> [Accessed: 15- Oct- 2017].
- [30] Linz Center of Mechatronics GmbH: *X2C Documentation*, Linz, 2016.