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 Square d 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 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 present among software engineers [5-7].

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.

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.

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**

<table>
<thead>
<tr>
<th>Gain Type</th>
<th>Speed</th>
<th>Current</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proportional</td>
<td>0.1084</td>
<td>8.1807</td>
</tr>
<tr>
<td>Integral</td>
<td>0.9453</td>
<td>12497.28</td>
</tr>
</tbody>
</table>

### 3 Simulation model

To support previous theoretical considerations, appropriate simulations were done in PLECS, using
a model shown in Fig. 2. The simulation model consisted of two main parts: a control subsystem presented in Fig. 2 (a) and a subsystem, shown in Fig. 2 (b), comprising models of an induction machine, a 3-phase inverter with six insulated-gate bipolar transistors (IGBTs), DC power supply and current, voltage, and speed measurements. Each simulation block in Fig. 2 performs a specific function and is built of several fundamental blocks available in PLECS libraries, with a detailed documentation in [23].

4 Experimental investigation

After the performed simulations, the control algorithm was implemented in X2C and applied to the actual system. This section provides a brief overview of the equipment and the testing methods employed.

4.1 Hardware setup

The experimental testing was conducted using a hardware setup that consisted of an induction machine, an incremental optical encoder, a DSP, an AC power supply and a laboratory motor control kit comprising a diode rectifier, DC link capacitors and a 3-phase IGBT inverter with measuring and protection devices. Described configuration is shown in Fig. 3. Although low power and with limited abilities, Marathon Electric’s induction machine [25] was appropriate for the testing purposes during the algorithm development phase. The machine nameplate data are given in Table 2. The dual data specified for the voltage and the current of the machine are related with the possibility of connecting the machine’s windings in two different ways, adapted for high and low voltage. During the machine operation, the connection with the nominal line voltage of 230 VAC (and therefore the DC link voltage of 325 VDC) was used.

### Table 2. Induction machine nameplate data [25]

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequency, $f_n$ (Hz)</td>
<td>60</td>
</tr>
<tr>
<td>Voltage, $U_n$ (V)</td>
<td>208-230/460</td>
</tr>
<tr>
<td>Number of phases</td>
<td>3</td>
</tr>
<tr>
<td>Mechanical output power, $P_n$ (W)</td>
<td>186</td>
</tr>
<tr>
<td>Speed at rated load, $n_{rn}$ (rpm)</td>
<td>1725</td>
</tr>
<tr>
<td>Rated load current, $I_n$ (A)</td>
<td>1.3-1.4/0.7</td>
</tr>
</tbody>
</table>

Electrical parameters of this machine, taken from [26], are given in Table 3. These parameters were used to calculate other parameters needed in the control structure, according to [19].

Mechanical parameters of the machine, including moment of inertia ($J$) and coefficients that model friction and windage losses ($k_2$, $k_1$, $k_0$), were obtained by two experiments.

Equation of motion of the induction machine is given by [19]:

\[ J \frac{d^2 \omega}{dt^2} + k_2 \omega + k_1 \frac{d\omega}{dt} + k_0 \omega = \frac{u - R_0 i}{\lambda_m} \]
\[
J \frac{d\omega}{dt} = T_e - T_{load} - T_{loss}
\]

where:
- \(J\) - moment of inertia (kgm\(^2\))
- \(\omega\) - 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]**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stator resistance, (R_s) ((\Omega))</td>
<td>11.05</td>
</tr>
<tr>
<td>Stator leakage inductance, (L_{sl}) (H)</td>
<td>0.02</td>
</tr>
<tr>
<td>Rotor resistance, (R_r) ((\Omega))</td>
<td>6.11</td>
</tr>
<tr>
<td>Rotor leakage inductance, (L_{rl}) (H)</td>
<td>0.02</td>
</tr>
<tr>
<td>Magnetizing inductance, (L_m) (H)</td>
<td>0.29</td>
</tr>
<tr>
<td>Rotor inductance, (L_r) (H)</td>
<td>0.31</td>
</tr>
<tr>
<td>Number of pole pairs, (p)</td>
<td>2</td>
</tr>
</tbody>
</table>

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

\[
\frac{d\omega}{dt} = 0
\]

Moreover, motor operation at no-load was considered:

\[
T_{load} = 0
\]

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:

\[
T_e = T_{loss}
\]

Developed electromagnetic torque is defined as [19]:

\[
T_e = \frac{3}{2} \rho \frac{L_m^2}{L_r} i_{mr} i_{sq}^2
\]

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

\[
T_{loss} = T_e = k_2 \omega^2 + k_1 \omega + k_0
\]

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**

<table>
<thead>
<tr>
<th>Angular speed (\omega) (rad/s)</th>
<th>Torque (T_e) (Nm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>18.064</td>
<td>0.00425</td>
</tr>
<tr>
<td>36.128</td>
<td>0.00900</td>
</tr>
<tr>
<td>54.192</td>
<td>0.01264</td>
</tr>
<tr>
<td>72.256</td>
<td>0.01548</td>
</tr>
<tr>
<td>90.320</td>
<td>0.01769</td>
</tr>
<tr>
<td>108.384</td>
<td>0.01979</td>
</tr>
<tr>
<td>126.448</td>
<td>0.02140</td>
</tr>
<tr>
<td>144.512</td>
<td>0.02304</td>
</tr>
<tr>
<td>162.576</td>
<td>0.02478</td>
</tr>
<tr>
<td>180.640</td>
<td>0.02586</td>
</tr>
</tbody>
</table>

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:

\[
r_i = T_i - \hat{T}_i
\]

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

\[
k = \left(W^T W\right)^{-1} W^T T
\]  

Finally, following values of coefficient estimates are obtained:

\[
k = \begin{bmatrix}
-6.000 \times 10^{-7} \\
2.372 \times 10^{-4} \\
8.310 \times 10^{-4}
\end{bmatrix}
\]  

The fitted values of torque are calculated as:

\[
\hat{T} = Wk
\]  

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 \(\omega_0\) 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.](image-url)
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:

\[ J \frac{d\omega}{dt} = -k_1\bar{\omega} \]  

(18)

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

Therefore, moment of inertia may be determined as:

\[ J = -\frac{k_1}{d\omega/dt} \frac{\bar{\omega}}{dt} \]  

(19)

Average speed \( \bar{\omega} \) was calculated as:

\[ \bar{\omega} = \frac{\omega(0s) + \omega(1s)}{2} = \frac{180 + 141.68}{2} = 160.84 \text{rad/s} \]  

(20)

Term \( d\omega / dt \) was obtained as a curve slope in the initial point \( \omega_0 \):

\[ \frac{d\omega}{dt}(\omega_0) = \frac{\Delta\omega}{\Delta t} = \frac{141.68 - 180}{1} = -38.32 \text{rad/s} \]  

(22)

Finally, moment of inertia was obtained:

\[ J = 9.828 \times 10^{-4} \text{kgm}^2 \]  

(24)

Estimated mechanical parameters are finally summarized in Table 5.

**Table 5. Induction machine mechanical parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Moment of inertia, ( J ) (kgm²)</td>
<td>9.828 \times 10^{-4}</td>
</tr>
<tr>
<td>Quadratic term coefficient, ( k_2 ) (Nms²)</td>
<td>-6.000 \times 10^{-7}</td>
</tr>
<tr>
<td>Linear term coefficient, ( k_1 ) (Nms)</td>
<td>2.372 \times 10^{-4}</td>
</tr>
<tr>
<td>Constant term, ( k_0 ) (Nm)</td>
<td>8.310 \times 10^{-4}</td>
</tr>
</tbody>
</table>

"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 µs. 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].

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]. 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

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.
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).

![Image](image_url)

(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:

\[
ISE = \int_{0}^{\infty} e^2(t)dt
\]

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...
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**

<table>
<thead>
<tr>
<th></th>
<th>Simulation</th>
<th>Experiment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum overshoot, $M_p$ (%)</td>
<td>3.14</td>
<td>3.36</td>
</tr>
<tr>
<td>Peak time, $t_p$ (s)</td>
<td>0.5295</td>
<td>0.5287</td>
</tr>
<tr>
<td>Settling time (2%), $t_s$ (s)</td>
<td>0.5900</td>
<td>0.5803</td>
</tr>
</tbody>
</table>

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**

<table>
<thead>
<tr>
<th></th>
<th>Simulation</th>
<th>Experiment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speed $n$</td>
<td>$5.2767\times10^2$</td>
<td>$5.3233\times10^2$</td>
</tr>
<tr>
<td>Direct stator current component $i_{sd}$</td>
<td>$1.6169\times10^{-5}$</td>
<td>$5.9533\times10^{-5}$</td>
</tr>
<tr>
<td>Quadrature stator current component $i_{sq}$</td>
<td>$1.9301\times10^{-4}$</td>
<td>$4.7905\times10^{-4}$</td>
</tr>
</tbody>
</table>

**5.2 Motor deceleration**

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

<table>
<thead>
<tr>
<th></th>
<th>ISE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speed $n$</td>
<td>$5.0753 \times 10^2$</td>
</tr>
<tr>
<td>Direct stator current component $i_{sd}$</td>
<td>$1.3259 \times 10^{-5}$</td>
</tr>
<tr>
<td>Quadrature stator current component $i_{sq}$</td>
<td>$3.9873 \times 10^{-4}$</td>
</tr>
</tbody>
</table>

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)).
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

<table>
<thead>
<tr>
<th></th>
<th>Simulation</th>
<th>Experiment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speed $n$</td>
<td>$1.5107 \times 10^3$</td>
<td>$1.0036 \times 10^3$</td>
</tr>
<tr>
<td>Direct stator current component $i_{sd}$</td>
<td>$5.0184 \times 10^{-4}$</td>
<td>$1.7805 \times 10^{-4}$</td>
</tr>
<tr>
<td>Quadrature stator current component $i_{sq}$</td>
<td>$0.8390 \times 10^{-3}$</td>
<td>$1.3178 \times 10^{-3}$</td>
</tr>
</tbody>
</table>

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


