Research Article

FPGA-Based Fixed Point Implementation of a Real-Time Induction Motor Emulator

L. Charaabi

L.S.E., Ecole Nationale d'Ingénieur de Tunis, BP 37, 1002 Le Belvédère, Tunisia

Correspondence should be addressed to L. Charaabi, lotfi.charaabi@enit.rnu.tn

Received 18 May 2012; Revised 12 September 2012; Accepted 18 September 2012

Academic Editor: Jose Pomilio

Copyright © 2012 L. Charaabi. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper investigates the numerical issue of a discrete-time induction-motor emulator implementation. The stability analysis of the finite-word-length implementation shows a coupling between required word length and the sample rate. We propose specific guidelines to analyze this coupling and to estimate the required data word length for both signals and coefficients of the model. To respect algorithm requirements, an FPGA-based implementation was used for architecture development. The direct torque control is implemented to verify in real time the AC-motor emulator prototype.

1. Introduction

Control algorithms of electrical drives are usually tested by the attachment of electrical motors. Experimental testing requires direct measuring by employing measurement instrumentation and sensors that would be complex, impractical, noise sensitive, and expensive. Traditionally these tests are performed on motor and inverter models under software simulation environments which are, in most cases, non-real-time and unable to exactly replicate real operational conditions.

In order to provide a real-time verification of the implemented control algorithm and increase the realism of simulation, the control algorithm is tested while connected to a real-time emulator for the plant (induction motor and inverter) [1]. Due to their high capacities of executing in real-time complex algorithm, FPGA technology is a good candidate for this kind of RT emulator [2].

For reasons of cost, simplicity, speeds and memory space, the real-time emulator of the induction motor is performed with the Euler's shift discretization method and quantized with fixed point format [3].

The robustness of the discretized algorithm is a critical issue in fixed-point format implementation. It is well known that, stable, closed-loop system may become unstable when the algorithm is implemented using a fixed-point processor

due to finite-word-length effects. The fractional part precisions in fixed point are chosen to guarantee a minimum signal-to-noise ratio for finite-word-length quantization effects. The integer part is computed using norms [4].

In an FPGA implementation, it is possible to use separate fixed-point format for each coefficient and signal in the algorithm. Hence, the use of FPGA allows maintaining a higher computational precision at critical points.

In this paper, authors propose a real-time induction motor emulator designed in fixed point format for an FPGA implementation. The starting point is a continuous-time model. Discrete-time models are derived using the traditional shift form approximation. A study in terms of stability and finite-world-length effects is shown and a design technique for choosing the coefficient and signal bit widths is given [5, 6].

To demonstrate the accurateness of the real-time induction motor emulator, a particular example is developed: the direct torque control (DTC) of an induction motor [7].

The paper is organized as follows. The second section reminds the algorithmic study of a discrete-time induction motor model-based shift form approximation. Section 3 gives an efficient methodology for choosing the finite fixed-point word length. Section 4 shows the FPGA-based architecture development and gives emulation results of the induction motor emulator with the DTC algorithm.

FIGURE 1: Block diagram of the DTC technique.

2. Discrete-Time Model Based on Shift Form Approximation

The block diagram of the complete system is outlined in Figure 1 including the control scheme, the induction motor, and the power chain.

As it can be seen, the DTC algorithm is decomposed in three specific blocks.

- (i) Transformation block, which is composed by two subalgorithms based on the direct Concordia transformation algorithm for the current and voltage three phases models.
- (ii) Estimation block, which is composed by the calculation subalgorithms of flux ($\phi_{s(d,q)}$, ϕ_s), torque (Γ_e), and flux sector position. The discrete-time model of the flux subalgorithm is based on the shift form approximation:

$$\phi_{s(d,q)}(k+1) = \phi_{s(d,q)}(k) + T_{vs(d,q)} - T i_{s(d,q)}. \quad (1)$$

- (iii) command block, which is composed by a look-up table (LUT) corresponding to the switching function of the power inverter.

In order to provide a real-time verification of the implemented control algorithm, the DTC is tested while connected to a real-time emulator for the induction motor and inverter.

The induction machine is a nonlinear high-order system and for this reason complicated models must be used to control it. The dynamic behaviour of the induction motors

can be described by a set of differential equations in a rotating reference frame with an angular velocity of ω_λ . Moreover it is assumed that a stationary reference frame is fixed to the stator axis. The proposed model also assumes the classical decoupling between mechanical and electrical modes. It yields that under these assumptions, the state model of the induction motor (IM) is [8]

$$\begin{aligned} \frac{d}{dt} y(t) &= A(\omega) y(t) + Bu(t), \\ y(t) &= Cx(t) + Du(t), \end{aligned} \quad (2)$$

where, $x = y = \begin{bmatrix} i_{sd} \\ i_{sq} \\ \phi_{rd} \\ \phi_{rq} \end{bmatrix}$, $u = \begin{bmatrix} u_{sd} \\ u_{sq} \end{bmatrix}$ matrices A , B , C , and D are 4-by-4, 4-by-2, 4-by-4, and 4-by-2, respectively:

$$A(\omega) = \begin{bmatrix} -\gamma & 0 & \frac{M_{sr}}{\sigma L_s L_r T_r} & \frac{M_{sr}}{\sigma L_s L_r} \omega \\ 0 & -\gamma & -\frac{M_{sr}}{\sigma L_s L_r} \omega & \frac{M_{sr}}{\sigma L_s L_r T_r} \\ \frac{M_{sr}}{T_r} & 0 & -\frac{1}{T_r} & -\omega \\ 0 & \frac{M_{sr}}{T_r} & \omega & -\frac{1}{T_r} \end{bmatrix},$$

$$B = \begin{bmatrix} \frac{1}{\sigma L_s} & 0 \\ 0 & \frac{1}{\sigma L_s} \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}. \quad (3)$$

$T_r = L_r/R_r$ is the rotor time constant, σ is leakage coefficient with $\sigma = 1 - M_{sr}^2/L_s L_r$, $\gamma = (R_s + M_{sr}/L_r T_r)/\sigma L_s$, and ω is the angular velocity.

By means of (2), the general equation of the electromagnetic torque is transformed to an advantageous form:

$$\Gamma_{em} = p \frac{M_{sr}}{L_r} (\phi_{rd} i_{sq} - \phi_{rq} i_{sd}). \quad (4)$$

The purpose of this section is to develop a computationally efficient discrete-time approximation of the continuous motor model operating in real time.

The most used discretization method is based on Forward shift approximation [8]. The shift form approximation is given by

$$\frac{d}{dt} = \frac{(z-1)}{T}, \quad (5)$$

where T is the sampling period.

Introducing (5) in (2) yields

$$\begin{aligned} x_z(k+1) &= A_z x_z(k) + B_z u(k), \\ y_z(k) &= C x_z(k) + D u(k), \end{aligned} \quad (6)$$

where

$$x_z(k) = y_z(k) = \begin{bmatrix} i_{sd}(k) \\ i_{sq}(k) \\ \phi_{rd}(k) \\ \phi_{rq}(k) \end{bmatrix}, \quad A_z = (I + AT), \quad B_z = BT. \quad (7)$$

The data flow graph (DFG) corresponding to the induction motor model is presented in Figure 2. Coefficients a_i depend on the parameters of the machine and are obtained for a per unit model.

The following lines show the different coefficients used for the per unit (PU) DTC algorithm

$$A = \begin{bmatrix} -a_1 & 0 & a_2 & a_3 \omega \\ 0 & -a_1 & -a_3 \omega & a_2 \\ a_5 & 0 & a_6 & -\omega \\ 0 & a_5 & \omega & -a_6 \end{bmatrix}, \quad (8)$$

where, $a_1 = \gamma$, $a_2 = (M_{sr}/\sigma L_s L_r T_r)(\phi_u/I_u)$, $a_3 = (M_{sr}/\sigma L_s L_r)(v_u/I_u)$, $a_4 = 1/\sigma L_s$, $a_5 = (M_{sr}/T_r)(I_u/\phi_u)$ and $a_6 = 1/T_r$.

The base values are determined from the maximal values by using the following equations, where I_m , v_m , and f_m are, respectively, the phase maximal current, the phase to

FIGURE 2: The DFG of induction motor model.

neutral maximal voltage, and the maximal frequency of the induction motor:

$$I_u = \sqrt{2} I_m, \quad v_u = \sqrt{2} v_m, \quad \phi_u = \frac{v_m}{2\pi f_m}. \quad (9)$$

The solution of the continuous system (2) is

$$x(t) = e^{At} x(0) + \int_0^t e^{A(t-\tau)} B u(\tau) d\tau. \quad (10)$$

And if we consider the solution at the instant kT , we obtain

$$x(kT) = (e^{AT})^k x(0) + \int_0^{kT} e^{A(kT-\tau)} B u(\tau) d\tau. \quad (11)$$

The solution at the instant kT to the system (6) is the following:

$$x_z(kT) = (I + AT)^k x_z(0) + T \sum_{l=0}^{k-1} (I + AT)^{k-l-1} B u(lT). \quad (12)$$

Therefore, the matrices that rule the continuous and discrete time systems are, respectively, e^{AT} and $(I + AT)$, and, consequently, the continuous and discrete time systems will be described by the eigenvalues or the poles of these matrices.

The poles of the matrix e^{AT} are $\lambda = e^{\beta T}$, β being a pole of the matrix A of the continuous time system; the poles of the matrix $(I + AT)$ are $(\lambda = 1 + \beta T)$.

FIGURE 3: Z-domain pole location of the motor model as function of the sampling frequency.

In this study, a low power induction motor (1.5 kW) has been considered. The corresponding continuous-time model has four poles for electromagnetic model as illustrated below:

$$\begin{aligned}\beta_1 &= (-46,86 + 0,5j)\omega + \frac{\sqrt{11336 - \omega^2 + 46i\omega}}{2}, \\ \beta_3 &= \overline{\beta_1}, \\ \beta_2 &= (-46,86 + 0,5j)\omega - \frac{\sqrt{11336 - \omega^2 + 46i\omega}}{2}, \\ \beta_4 &= \overline{\beta_2}.\end{aligned}\quad (13)$$

Let us suppose the following quadratic error function:

$$\varepsilon = (e^{\beta T} - 1 - \beta T)(e^{\beta T} - 1 - \beta T)^*. \quad (14)$$

Therefore, for sampling periods T tending to zero the difference between $e^{\beta T}$ and $(1 + \beta T)$ becomes very small for each β_i pole of A , and, consequently, it will optimize the resemblance between the time-domain response of the continuous-time and the discrete-time system.

Discrete-time systems resulting from the shift form approximation are sometime unstable. It is possible, however, to select the sampling rate such that the discrete-time system is always stable when the corresponding continuous-time system is stable. For the shift form realization, the stability domain is located inside the unit circle. Therefore, a z -domain pole is stable if its module is less than one:

$$|\beta_i T + 1| < 1. \quad (15)$$

Therefore, the most critical z -domain pole corresponds to the s -domain pole having the greatest module. The pole β_1 has the greatest module as function of ω .

The locations of poles of the motor for different sampling rate values are shown in Figure 3. It can be seen that for high sampling rate, the poles become very close to one. This can be problematic, particularly when finite precision formats are considered.

3. Efficient Methodology for Choosing the Word Length of Coefficients and Variables

The following methodology is developed for choosing each coefficient and variable bit width.

3.1. Representation of Coefficients. When the model is implemented in fixed-point format, each coefficient a_k must be given a finite-precision representation composed by an integer and fractional part. Each part is estimated separately.

The integer part position for a coefficient a_k must be estimated by taking the base 2 logarithm of the maximum coefficient:

$$B_{\text{int}} = \lfloor \log_2 |a_k| \rfloor + 1, \quad (16)$$

where $\lfloor x \rfloor$ rounds x to the nearest integer less than or equal to x .

The fractional part must be determined by the maximum allowable perturbation of the coefficient from its ideal infinite-precision value. This can be achieved by computing the quantized parameter deviation or sensitivity and the stability margin. The angular velocity is selected to be the worst-case value. By means of Figure 3, for pole λ_1 and its conjugate, ω is fixed to the maximal value and for pole λ_2 and its conjugate ω is equal to zero.

Let $\{\hat{a}_k\}$ be the quantized parameter set consisting of all quantized coefficients of the state matrix A_z . When the model is implemented in fixed-point format, each coefficient of the $\{a_k\}$ set is perturbed:

$$\{\Delta a_k = \hat{a}_k - a_k\}. \quad (17)$$

Due to the finite-word-length effects, each Δa_k perturbation is bounded by

$$|\Delta a_k| \leq 2^{-B_F}, \quad (18)$$

where B_F is the number of bits of the fractional part.

Due to the perturbation $\{\Delta a_k\}$ each pole $\lambda_i[(a_k)]$ is moved to $\hat{\lambda}_i = \lambda_i[A_z(a_k + \Delta a_k)]$ and it follows from a first-order approximation that the deviation can be computed with

$$\Delta \lambda_i = \hat{\lambda}_i - \lambda_i = \sum_{k=1}^N \frac{\partial \lambda_i}{\partial a_k} \Delta a_k, \quad (19)$$

where N is the number of modified parameters ($\Delta a_k \neq 0$).

As an example, for poles λ_2 and λ_4 , the worst-case deviation is obtained with $\omega = 0$. Then, the expression of deviation

$$\Delta \lambda_2 = \Delta \lambda_4 = \frac{\partial \lambda_2}{\partial a_1} \Delta a_1 + \frac{\partial \lambda_2}{\partial a_2} \Delta a_2 + \frac{\partial \lambda_2}{\partial a_5} \Delta a_5 + \frac{\partial \lambda_2}{\partial a_6} \Delta a_6 \quad (20)$$

with, $\partial\lambda_2/\partial a_1 = -\partial\lambda_2/\partial a_6 = -(T/2)(1 + (a_1 - a_6)/\sqrt{(a_1 - a_6)^2 + 4a_5a_2})$ and $\partial\lambda_2/\partial a_2 = (a_2/a_5)(\partial\lambda_2/\partial a_5) = -(T/2)(2a_5/\sqrt{(a_1 - a_6)^2 + 4a_5a_2})$.

The quantized pole $\hat{\lambda}_i$ may be outside the unit circle, especially with high sampling rate; thus, it is critical to know when the finite-word-length error leads to instability. This means to determine how close $\lambda_i[(a_k)]$ are to the unit circle. Let us consider the following related measure for each pole:

$$\left| \frac{\sum_{k=1}^N (\partial\lambda_i/\partial a_k) \Delta a_k}{1 - |\lambda_i(a_k)|} \right| < \varepsilon, \quad (21)$$

where $1 - |\lambda_i(a_k)|$ is called the stability margin of the i th pole and ε is a prescribed maximum allowable percent change in pole location relative to the stability margin.

Assuming that all coefficients have the same word-length fractional part, then $\Delta a_k = \Delta a \ \forall k \in \{1, \dots, N\}$; it follows from inequality (21) that

$$|\Delta a| < \frac{|1 - |\lambda_i(a_k)||}{\left| \sum_{k=1}^N (\partial\lambda_i/\partial a_k) \right|} \varepsilon. \quad (22)$$

From inequalities (22) and (18), the required fractional part word length can be derived as

$$B_F = \left\lceil \log_2 \left(\frac{\left| \sum_{k=1}^N (\partial\lambda_i/\partial a_k) \right|}{|1 - |\lambda_i(a_k)||} \frac{1}{\varepsilon} \right) \right\rceil + 1. \quad (23)$$

For the studied motor, setting $\varepsilon = 5\%$ and $T = 100 \mu s$, the conditions (23) yield 19 bits for fractional part word length for A matrix coefficients.

3.2. Representation of Signals. The next step in the fixed-point motor model implementation is to estimate the format for each signal. An incremental methodology is used to determine the appropriate representation for the state variables. As shown in Figure 4, the first step is to find transfer function from each input and the appropriate signal in the DFG. Then the dynamic range (maximum amplitude) of each signal is carried out by using l_∞ and l_1 norms, and, thus, the integer word length range is identified. Finally, the fractional part word-length is estimated by analysis of the signal-to-quantization-noise ratio (SQNR) for each signal.

The dynamic range (maximum amplitude) of each signal of the DFG allows identifying their integer range. In practice the maximal current supported by the motor as well as the maximal produced torque and fluxes is known. Therefore, for the per-unit motor model, these dynamic ranges are bounded by 1. Notice that the voltage inputs and the electrical velocity are also normalized. Thus, these dynamic ranges are all bounded by 1:

$$\begin{aligned} \|i_{sd}\|_\infty &= \|i_{sq}\|_\infty = \|\Gamma_{em}\|_\infty = \|v_{sq}\|_\infty = \|v_{sd}\|_\infty \\ &= \|\omega\|_\infty = \|\phi_{rd}\|_\infty = \|\phi_{rq}\|_\infty = 1, \end{aligned} \quad (24)$$

where $\|\cdot\|_\infty$ denotes the l_∞ norm, defined by

$$\|x\|_\infty = \max_k |x(k)|. \quad (25)$$

FIGURE 4: Incremental methodology for signal representation.

TABLE 1: Range bounds for the motor model signals.

Signal	Range bound	Integer size
$v_{sd}, v_{sq}, \Gamma_{em}, \omega, i_{sd}, i_{sq}, \phi_{rd}, \phi_{rq}$	1	1
y_1, y_2	1.03	2

Now considering internal y_1 and y_2 signals of the DFG, one can see that

$$\begin{aligned} y_1(z) &= a_4 v_{sd}(z) + a_3 \phi_{rd}(z) \omega(z) + a_1 i_{sd}, \\ y_2(z) &= a_4 v_{sq}(z) + a_3 \phi_{rq}(z) \omega(z) + a_1 i_{sq}. \end{aligned} \quad (26)$$

Therefore, signals y_1 and y_2 are bounded by the same quantity:

$$\|y_1\|_\infty = \|y_2\|_\infty \leq |a_4| + |a_1| + |a_2|. \quad (27)$$

The integer part word length of all the signals is then estimated by taking the base 2 logarithm of the range bound. Table 1 illustrates the signal range bounds and the corresponding integer part word length.

The fractional part word length is estimated by analyzing the signal-to-quantization-noise ratio (SQNR) for each signal [9]. The SQNR is measured in decibels as

$$\text{SQNR} = 10 \log_{10} \left(\frac{\sigma_{\text{signal}}^2}{\sigma_{\text{noise}}^2} \right). \quad (28)$$

There are two possible sources of quantization noise in the induction motor emulator implementation, namely:

- the noise introduced by the limited resolution of a possible control technique, producing an input noise variance σ_i^2 ,
- the truncation noise which is introduced when products or sums are quantized. This noise propagates through the emulator and appears at outputs.

It is assumed that the quantization noise has the following properties.

- (i) Each quantization noise source is a stationary white noise process.
- (ii) Each noise source is uncorrelated with all other noise sources and the input of the system.
- (iii) The error resulting from quantization can then be modeled as a random variable uniformly distributed over the appropriate error range. Therefore, the noise variance due to truncation is

$$\sigma_t^2 = \frac{2^{-2B}}{12}, \quad (29)$$

where B is the fractional part word length.

To investigate the noise propagation, a noise quantization model of the induction motor emulator is developed in Figure 5. The addition is performed without truncation; the quantization is performed after each multiplication; the quantization noise resulting from N multiplications is therefore N -times higher than the one generated by a single multiplication. It yields that the variances of noise e_{t1} and e_{t2} are

$$\sigma_{t1}^2 = \sigma_{t4}^2 = 4 \frac{2^{-2B_a}}{12}, \quad \sigma_{t2}^2 = 3 \frac{2^{-2B_b}}{12}. \quad (30)$$

The output noise variances can be estimated using the matrix transfer function from the input vector $u(k) = [e_{t1}(k), e_{t1}(k), e_{t2}(k), e_{t2}(k)]$ to the output vector $y(k) = [f_i(k), f_i(k), f_\phi(k), f_\phi(k)]$. Using matrix notation in (5), the transfer function matrix response is

$$\begin{aligned} H_{v \rightarrow \phi, i}(z) &= D' + C(zI - A_z)^{-1}B' \\ &= \begin{bmatrix} H_{11}(z) & H_{12}(z) & H_{13}(z) & H_{14}(z) \\ -H_{12}(z) & H_{11}(z) & -H_{14}(z) & H_{13}(z) \\ H_{31}(z) & H_{32}(z) & H_{33}(z) & H_{34}(z) \\ -H_{32}(z) & H_{31}(z) & -H_{34}(z) & H_{33}(z) \end{bmatrix}, \end{aligned} \quad (31)$$

where

$$B' = \begin{bmatrix} a_4 & 0 & 0 & 0 \\ 0 & a_4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad D' = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \quad (32)$$

Let $h_{x \rightarrow y}(k)$ denote the impulse response from input x to output y for a linear system. Assuming that the input signal has a maximal value $x_{\max}$, the output y must be bounded [10] so that

$$\|y\|_\infty \leq x_{\max} \|h_{x \rightarrow y}\|_1, \quad (33)$$

where $\|\cdot\|_1$ denotes the l_1 norm, defined by $\|x\|_1 = \sum_{k=0}^{\infty} |x(k)|$.

FIGURE 5: An additive noise model for the truncation and control technique noise through the induction motor model.

Using (31) and (33), the variances of resulting output noises are governed by these bounds [11]:

$$\begin{aligned} \|\sigma_{f_i}^2\|_\infty &\leq (\sigma_{t1}^2 + \sigma_i^2)(\|H_{11}(z)\|_1 + \|H_{12}(z)\|_1) \\ &\quad + (\sigma_{t2}^2)(\|H_{13}(z)\|_1 + \|H_{14}(z)\|_1), \\ \|\sigma_{f_\phi}^2\|_\infty &\leq (\sigma_{t1}^2 + \sigma_i^2)(\|H_{31}(z)\|_1 + \|H_{32}(z)\|_1) \\ &\quad + (\sigma_{t2}^2)(\|H_{33}(z)\|_1 + \|H_{34}(z)\|_1). \end{aligned} \quad (34)$$

Each coefficient $\|H_{ij}(z)\|_1$ is computed to have a maximal value as function of electrical velocity ω :

$$\|H_{ij}(z)\|_1 = \max_\omega \|H_{ij}(z, \omega)\|_1. \quad (35)$$

The l_1 norms of discrete-time impulse responses corresponding to these transfer functions are computed approximately by finite sums. Figure 6 shows the impulse response for each coefficient of the matrix $H_{v \rightarrow \phi, i}(z)$.

$$\|\sigma_{f_i}^2\|_\infty \leq 9697 \frac{2^{-2B}}{12} + 1246(\sigma_i^2), \quad (36)$$

$$\|\sigma_{f_\phi}^2\|_\infty \leq 2275 \frac{2^{-2B}}{12} + 115(\sigma_i^2). \quad (37)$$

Assuming that all signals have the same word length, the first inequality gives the worst-case variance error and

FIGURE 6: Impulse response of the matrix $H_{v \rightarrow \phi, i}$ with sample period $T = 100 \mu s$.

therefore the fractional part word length will be selected taking into account this condition.

The variances of the sinusoidal input and output signals of the induction motor emulator are expressed in terms of the root mean square. Hence (18) gives

$$\sigma_{\text{signal}}^2 = \text{signal}_{\text{rms}}^2 = \frac{1}{2}. \quad (38)$$

Assume that

$$\sigma_i^2 = \frac{2^{-2B_i}}{12}, \quad (39)$$

where B_i is the fractional part word length of the input voltage. The resolution is strongly dependent on the PWM one.

Substituting (38) and (28) in (36) gives this condition:

$$\text{SQNR}_{f_i} \geq -10 \log_{10} (1628.2^{-2B} + 208.2^{-2B_i}). \quad (40)$$

Figure 7 shows the SQNR as function of the fractional part word length for different B_i bits. With a 10-bit B_i wordlength, we obtain $\text{SQNR} = 75 \text{ dB}$ for 18-bit B wordlength.

4. FPGA-Based Induction Motor Emulator

4.1. System Architecture. The induction motor emulator is implemented using modular and standard design principles on a Xilinx Development kit, which contain a FPGA from SPARTAN III family, the XC3s200.

FIGURE 7: SQNR as function of the signal fractional part word length.

The emulator module requires important hardware resource. Therefore, the corresponding architecture is optimized in terms of consumed resources by the AAA methodology [12] which leads to only 3 multipliers.

As in [13], the proposed architecture consists in a control unit and a data path. The control unit generates a suitable timing schedule to control the data path and other local sequencers. Figure 8 depicts the proposed system architecture.

FIGURE 12: Real-time simulation results, obtained with DTC algorithm and the IM emulator: (a) steady-state stator flux vector locus, (b) estimated torque response to a step control from 0 N·m to 4 N·m. $T = 100 \mu\text{s}$, $\omega = 500 \text{ rpm}$ and $\Phi_s^* = 1.15 \text{ wb}$.

- (iii) The IM module that requires the most important hardware resources. It implements the induction motor model.
- (iv) The global sequencer module. It represents the control unit that generates a suitable timing schedule to control data path and other local sequencers.

The real-time simulation results presented in Figure 10 with 10 kHz sampling frequency show a slight difference between the torque produced by the implemented motor model architecture and the simulated torque under MATLAB environment under continuous-time conditions. This difference is too small to affect the performance of the real-time model.

4.3. System Architecture under Test with DTC Algorithm. The DTC is a well-known induction motor control strategy [7]. This control algorithm is selected to be tested with the real time induction motor model. The implemented architecture of the DTC control algorithm has been tested with the induction motor model architecture in real time. Figure 11 illustrates the proposed real time test bed. It includes a set of dedicated modules to guarantee a real-time simulation.

- (i) The UART (Universal Asynchronous Receiver Transmitter) module that provides a serial communication between the host PC and the implemented architecture.
- (ii) The stimuli module. It generates the references for the DTC control algorithm.

The torque and the stator flux are collected from the serial interface and visualized under Matlab-Simulink environment. Figure 12 illustrates the real-time emulation results.

5. Conclusions

This work has presented an optimized fixed-point format induction motor model intended for real-time simulation and emulation. With the analysis of the coupling between the sample rate and the data word length, this work has provided a theoretical guideline to find the optimal hardware implementation. The proposed architecture of the model has been successfully verified by the development and implementation of a real-time test bed that contains the AC-motor model and a DTC control algorithm.

The methodology for estimating the appropriate data word length may be applied to other various discrete-time system realizations.

Acknowledgment

This paper was supported by the Tunisian Ministry of High Education and Research: UR-LSE-ENIT-03/UR/ES05.

References

- [1] S. Lentijo, A. Monti, E. Santi, C. Welch, and R. Dougal, "A new testing tool for power electronic digital control," in *Proceedings of the IEEE 34th Annual Power Electronics Specialists Conference*, vol. 1, pp. 81–87, Acapulco, Mexico, June 2003.
- [2] T. Grandpierre, C. Lavarenne, and Y. Sorel, "Optimized rapid prototyping for real-time embedded heterogeneous multiprocessors," in *Proceedings of the 1999 7th International Conference on Hardware/Software Codesign (CODES '99)*, pp. 74–78, Rome, Italy, May 1999.
- [3] J. Wu, S. Chen, G. Li, and J. Chu, "Optimal finite-precision state-estimate feedback controller realizations of discrete-time systems," *IEEE Transactions on Automatic Control*, vol. 45, no. 8, pp. 1550–1554, 2000.

- [4] D. Menard and O. Sentieys, "Automatic evaluation of the accuracy of fixed-point algorithms," in *Proceedings of the IEEE/ACM Conference on Design, Automation and Test in Europe (DATE '02)*, Paris, France, March 2002.
- [5] H.-M. Cheng and G. T.-C. Chiu, "Coupling between sample rate and required wordlength for finite precision controller implementation with delta transform," in *Proceedings of the American Control Conference (ACC '07)*, pp. 3588–3593, New York, NY, USA, July 2007.
- [6] R. S. H. Istepanian, G. Li, J. Wu, and J. Chu, "Analysis of sensitivity measures of finite-precision digital controller structures with closed-loop stability bounds," *IEEE Proceedings of control Theory Application*, vol. 145, no. 5, pp. 472–478, 1998.
- [7] I. Takahachi and T. Nogushi, "A new quick response and high efficiency control strategy of an induction motor," *IEEE Transactions on Industry Applications*, vol. IA-22, no. 5, pp. 820–827, 1986.
- [8] O. Vainio, S. J. Ovaska, and J. J. Pasanen, "A digital signal processing approach to real-time AC motor modeling," *IEEE Transactions on Industrial Electronics*, vol. 39, no. 1, pp. 36–45, 1992.
- [9] L. Charaabi, E. Monmasson, and I. Slama-Belkhodja, "FPGA-based real-time emulation of induction motor using fixed point representation," in *Proceedings of the 34th Annual Conference of the IEEE Industrial Electronics Society (IECON '08)*, pp. 2393–2398, November 2008.
- [10] Z. Fang, J. E. Carletta, and R. J. Veillette, "A methodology for FPGA-based control implementation," *IEEE Transactions on Control Systems Technology*, vol. 13, no. 6, pp. 977–987, 2005.
- [11] M. H. Hayes, *Schaum's Outline of Theory and Problems of Digital Signal Processing*, McGraw-Hill, New York, NY, USA, 1998.
- [12] T. Grandpierre, C. Lavarenne, and Y. Sorel, "Optimized rapid prototyping for real-time embedded heterogeneous multiprocessors," in *Proceedings of the 1999 7th International Conference on Hardware/Software Codesign (CODES'99)*, pp. 74–78, May 1999.
- [13] E. Monmasson and M. N. Cirstea, "FPGA design methodology for industrial control systems-a review," *IEEE Transactions on Industrial Electronics*, vol. 54, no. 4, pp. 1824–1842, 2007.